VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2017 **Ondřej Michálek**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Realizace výukové úlohy z oblasti automatického řízení na platformě REXduino

Implementation of Educational Model of Control System using REXduino Platform

2017 Ondřej Michálek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Ondřej Michálek**

Studijní program: B2649 Elektrotechnika

Studijní obor: 2612R041 Řídicí a informační systémy

Téma: Realizace výukové úlohy z oblasti automatického řízení na platformě REXduino
Implementation of Educational Model of Control System using REXduino Platform

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznámení se s počítačem Raspberry Pi a s mikrokontrolérem Arduino Uno.
2. Seznámení se s řídicím systémem REX se zaměřením na danou platformu.
3. Dokumentace testovacího výukového stojanu využívajícího vstupně výstupní signály mikrokontroléru Arduino Uno z hlediska řízení, návrh a softwarové řešení úlohy, vizualizace.
4. Implementace regulačního obvodu pro zvolenou regulovanou soustavu s využitím testovacího výukového stojanu, návrh a softwarové řešení úlohy, vizualizace.
5. Zhodnocení dosažených výsledků závěrečné práce, závěr.

Seznam doporučené odborné literatury:

[1] XUE, Dingyü, Yangquan CHEN a Derek P. ATHERTON. *Linear feedback control: analysis and design with MATLAB.* 4th rev. and enlarged ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, c2007. xii, 354 p. ISBN 08-987-1638-1/978-0-898716-38-2.
[2] NOSKIEVIČ, Petr. *Modelování a identifikace systémů.* 1. vyd. Ostrava: MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.
[3] VÍTEČKOVÁ, Miluše a Antonín VÍTEČEK. *Základy automatické regulace.* 2. vyd. Ostrava: VŠB-TU Ostrava, 2008. 243 s. ISBN 978-80-248-1924-2.
[4] VAVŘÍN, Petr. *Teorie automatického řízení I (Lineární spojité a diskrétní systémy).* 2. přepracované vyd. Brno: VUT Brno, 1991. 158 s. ISBN 80-214-0244-X.
[5] VAVŘÍN, Petr. *Teorie dynamických systémů.* 1. vyd. Brno: VUT Brno, 1989. 177 s.
[6] ZÍTEK, Pavel, Milan HOFREITER a Jaroslav HLAVA. *Automatické řízení.* Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 1999. 148 s. ISBN 80-010-2044-4.
[7] ŠULC, Bohumil. *Teorie automatického řízení s počítačovou podporou.* Vyd. 1. Praha: ČVUT, Strojní fakulta, 1999. 154 s. ISBN 80-010-1974-8.
[8] OŽANA, Štěpán. *Navrhování a realizace regulátorů.* vyd. 1. Ostrava: VŠB - TU Ostrava, 2012. Studijní materiály. 136 s. ISBN 978-80-248-2605-9.
[9] ZEZULKA, František, Petr FIEDLER a Zdeněk BRADÁČ. *Prostředky průmyslové automatizace.* Učební texty. Brno: VUT v Brně, 2002.
[10] Firemní dokumentace Matlab and Simulink (MathWorks).
[11] Firemní dokumentace Rex Controls.
[12] Firemní dokumentace Arduino Uno.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Štěpán Ožana, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

„Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.“

V Ostravě dne 27.4.2017 ……

Ondřej Michálek

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce doc. Ing. Štěpánovi Ožanovi, Ph.D. za pomoc, čas obětovaný konzultacím a radám, které mi hodně pomohly. Dále bych rád poděkoval panu Ing. Jakubovi Noskovi za odborné rady a pomoc s identifikací systému fyzikální úlohy a návrhem PID regulátoru.

Abstrakt

Předmětem bakalářské práce „Realizace výukové úlohy z oblasti automatického řízení na platformě „REXduino“ je seznámení s mikrokontrolerem Arduino UNO s počítačem Raspberry Pi, ve spojení s řídicím systémem REX, který umožňuje dokonale propojení těchto dvou zařízení. Dalším cílem je vytvoření modelu regulátoru pro vybranou fyzikální úlohu a aplikování jej na výukovém testovacím stojanu pro praktické znázornění regulačních obvodů ve výuce předmětu Kybernetika. Do této problematiky spadá jednak identifikace systému a jednak i návrh vhodného regulátoru. Na závěr je vytvořena vizualizace regulované soustavy programem RexHMI Designer pro podporu uživatelského rozhraní.

Klíčová slova

Arduino UNO R3, Raspberry Pi 3, REXduino, řídící systém REX, identifikace, vizualizace, HMI, RexHMI Designer, regulace, syntéza regulátoru, PID regulátor, fyzikální úloha, RC článek, funkční bloky, Matlab.

Abstract

The subject of this thesis is to introduce with the microcontroller Arduino UNO and computer Raspberry Pi, which are controlled by the control system REX. The platform created by combining these two devices is called REXduino. The control systém REX allow perfect connection of these two devices. The next target is to create a model of the controller to the selected physical task and applying it to practical teaching model for representation of control circuits in teaching the subject cybernetics. This problem is solved by the system identification and design of a suitable controller. At the end designed HMI for controlled system by the RexHMI Designer.

Key Words

Arduino UNO R3, Raspberry Pi 3, REXduino, control system REX, identification, visualization, HMI, RexHMI Designer, regulation, synthesis of the controller, PID controller, physical task, RC circuit, function blocks, Matlab.

Obsah

Seznam použitých symbolů a zkratek

A_0, A_1, A_2		koeficienty optimálního modulu
a_0, a_1, a_2, a_3		koeficient charakteristického polynomu
B_0, B_1, B_2		koeficienty optimálního modulu
$b_0, b_1,$		koeficient charakteristického polynomu
C	[F]	elektrická kapacita
$e_{(0)}$		regulační odchylka
GPIO		general-purpose input/output
f	[Hz]	frekvence
$G_{(o)}$		přenos otevřeného regulačního obvodu
$G_{(s)}$		přenos soustavy
$G_{(W)}$		přenos uzavřeného regulačního obvodu
I	[A]	elektrický proud
k		činitel zesílení
l	[m]	délka
PID		proporcionálně, integrační, derivační

PWM		pulse Width Modulation, pulzně šířková modulace
R	[Ω]	elektrický odpor
SVG		grafický formát
TCP/IP		sada protokolů pro komunikaci v počítačové síti
T	[s]	časová konstanta
T_d	[s]	derivační konstanta
T_i	[s]	integrační konstanta
T_u	[s]	doba průtahu
T_n	[s]	doba náběhu
U	[V]	elektrické napětí

Seznam obrázků a seznam tabulek

1 Úvod

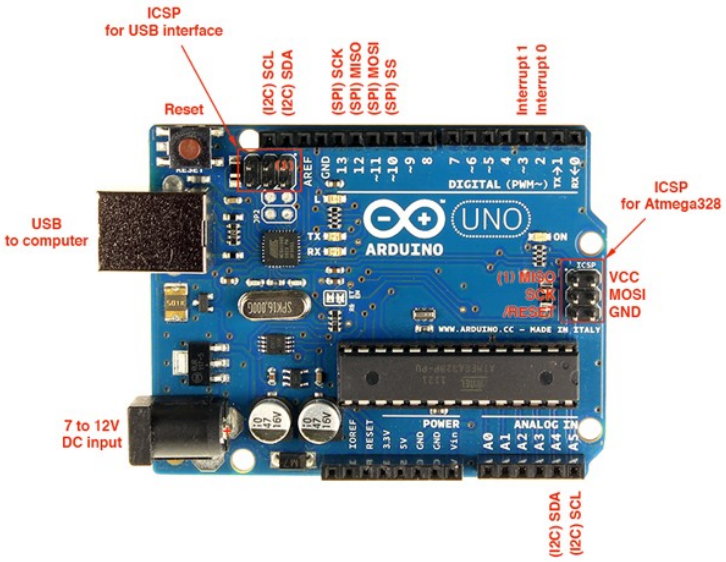
Řízení a regulace je neodmyslitelnou součástí dnešního průmyslu, ať se jedná o řízení spojité, číslicové či diskrétní. V dnešní době si život bez automatizace nelze představit. Zejména v momentech, kdy je třeba zamyslet se nad energetickými, lidskými, časovými, či jinými úsporami ve výrobě nebo v případě požadavku na dodržení kvality a stálosti procesu. Obrovský rozvoj řídicích systémů v poslední době umožňuje při návrhu řízení pro danou aplikaci nespočetně možností a je vhodné již na počátku stanovit požadavky na dané zařízení. Při nevhodném návrhu může dojít ke zbytečnému navýšení rozpočtu projektu či neefektivnímu řízení automatizovaného procesu. V tomto momentu, kdy je potřeba navržení správného řídícího algoritmu je ideální znát detailní popis chování systému.

Hlavním cílem této práce je aplikování regulačního obvodu řízeného platformou REXduino na zvolenou regulovanou soustavu fyzikální úlohy, která bude použita jako demonstrační model do předmětu kybernetika. Platforma REXduino se skládá z mikrokontroleru Arduino a minipočítače Raspberry Pi, řízené řídícím systém REX. Tímto spojením vzniká levná, avšak výkonná platforma pro implementaci řídících systémů. Práce je rozdělena na teoretickou část a praktickou část. V teoretické části čtenáře seznámím s jednotlivými zařízeními a samotným systémem REX, který regulovanou soustavu bude řídit. Dále popíšu způsoby, kterými lze identifikovat systémy a navrhnout regulátor. V praktické části poté aplikuji jednotlivé metody návrhu regulátoru pro moji zvolenou soustavu a v druhé polovině je ukázka softwaru a popis řízení s vizualizací.

Problematika vizualizace řeší rozhraní mezi operátorem a řídícím procesem (HMI) a zprostředkovává přenos informací. Na implementovaný regulační obvod je tedy vytvořená uživatelská vizualizace vykreslená vektorovou grafikou grafickým editorem Inkscape. Ten je součástí nove verze REX, pro snadnější práci. Vizualizaci můžeme zobrazit pomocí nových internetových prohlížečů podporující HTML5, kdy se připojíme k platformě REXduino přes lokální nebo internetovou síť.

2 Teoretická část

2.1 Arduino



Obr. 1: Arduino UNO R3 [1]

Arduino (Obr. 1) je malý mikrokontrolér založený na mikroprocesorech od firmy Atmel. Původně navrhnutý jako vývojový set pro studenty, kteří si nechtěli pořizovat v té době drahé mikrokontroléry. Arduino se stalo mezi studenty velmi oblíbený, a tak se tvůrci rozhodli jej poskytnout celému světu pro kutily, „bastlíře“, umělce, designéry a pro každého, koho zajímá vytváření interaktivních objektů. Arduino je Open Source projekt, což znamená, že jsou dostupné všechny dokumenty, schémata a návody. Je možné jej tedy i složit ručně, vylepšit jej nebo koupit složené a otestované.

Arduino vnímá okolí prostřednictvím vstupů z rozličných senzorů a zároveň jej může ovlivňovat připojenými pohony, LED diodami, LCD displeji a dalšími výstupními periferiemi. Nebavíme se tedy o počítači jako jsou běžně známé stolní počítače. Nelze k němu připojit přímo monitor ani klávesnici či myš. Mikroprocesor na desce se programuje pomocí speciálního programovacího ve vlastním vývojovém prostředí Arduino IDE, které je napsané v jazyce Java. Je založen na výukovém prostředí Processing a Wiring, který je podobný programovacímu jazyku C.

Na rozdíl Raspberry Pi není Arduino zamýšleno jako plnohodnotný stolní počítač. Řídicí program je vyvíjen na stolním počítači a do Arduina je poté nahrán a spuštěn. Na Arduinu běží pouze nahraný program, který probíhá v klasické, neustále se opakující smyčce. Tím se neustále kontroluje stav okolí a reaguje se okamžitě na změny. Podobně funguje i princip PLC automatů.

Pro moji práci jsem zvolil Arduino UNO R3 o rozměrech 69 x 53 mm. Tento mikrokontrolér je založen na mikroprocesoru ATmega328 o frekvenci krystalu 16 MHz. Obsahuje 14 digitálních vstupních i výstupních pinů, z toho 6 z nich může být použito jako výstupy pro PWM a 6 analogových vstupů. Na desce je k dispozici také RESET tlačítko, indikační LED diody, ICSP rozhraní či napájecí konektor. Je to v současnosti nejčastěji používaný typ desky a propojení se stolním počítačem je možné pomocí klasického USB portu. [2][3][4]

2.1.1 Specifikace

- Mikroprocesor: Atmega328
- Frekvence: 16 MHz
- Paměť: 32 kB
- SRAM: 2 kB
- EEPROM: 1 kB
- Vstupy a výstupy: 14x digitální vstup a výstup (sériový port, externí přerušení, PWM), 6x analogový vstup
- Rozhraní: USB 2.0 (převodník na sériový port), SPI konektor
- Vstupní napětí: 7-12 V
- Maximální vstupní napětí: 6-20 V
- Pracovní napětí: 5 V
- DC proud na pin: 40 mA
- Rozměry: 6,85 x 5,33 cm

2.2 Raspberry Pi

Raspberry Pi (Obr. 2) je malý jednodeskový počítač za rozumnou cenu o rozměrech 85,6 mm x 56,5 mm, který byl vyvinut původně pro výuku informatiky na školách, za účelem seznámení studentů s řízením počítačů na různých zařízení a vzbuzením jejich zájmu k programování. Vizí návrhářů bylo vytvoření snadno programovatelného zařízení s dostupnými periferiemi, tak aby se studenti naučili vytvářet software, ale aby pochopili i hardware a porozuměli funkci počítače.

Původní záměr tvůrců v zatraktivnění výuky byl splněn, ale postupem času se Raspberry Pi stalo natolik populární, že se kolem tohoto malého počítače vytvořila velká komunita technických nadšenců, kteří Raspberry Pi začali využívat jako univerzální stroj pro řešení celé řady aplikací. Od jednoduchých kutilských aplikací s elektronikou po domácí automatizaci, multimediální centra, ovládání dronů, robotů nebo řízení jednoduchých průmyslových aplikacích. Možností využití je celá škála, díky univerzálnosti a otevřeností linuxového prostředí.

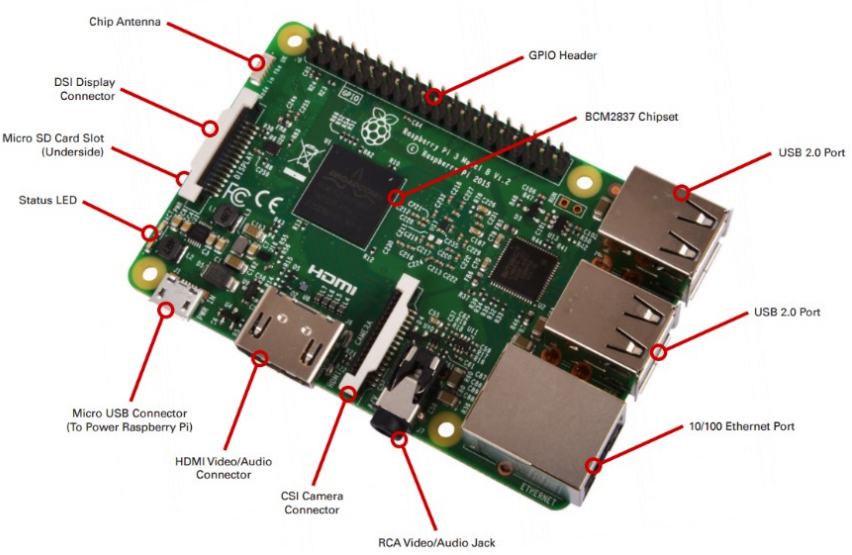
Od prvního modelu uběhla dlouhá doba a vývojáři stále pracují na zdokonalování, proto již vyšla spousta nových modelů. Posledním modelem je Raspberry Pi 3 model B. Tento model byl i vybrán

pro moji práci. Od původního modelu došlo k rozšíření počtu USB portů, výkonu CPU, velikosti paměti, výkonu grafické karty s podporou HDMI, či komunikaci přes Bluetooth a Wifi. Vzhledem k rozvoji výpočetní techniky a popularitě zařízení, lze očekávat v budoucnu další modely.

Raspberry Pi 3 je model třetí generace s 1,2GHz 64-bitovým procesorem ARM, 1GB RAM a integrovanou WiFi (b/g/n) a Bluetooth 4.1 Zachovává si stejné rozměry a rozmístění konektorů jako předchozí modely. Oproti předchozímu modelu má o 33 % výkonější procesor a modernější jádro, které zvládá lépe aplikace s 64-bitovými hodnotami. Grafický výkon také vzrostl, frekvence VideoCore se zvýšila na 400MHz pro zpracování 3D grafiky. Při pohledu na Raspberry Pi 3 je celkově o 50 % rychlejší než jeho předchozí model. Největší změnou je ale integrace WiFi 802.11 b/g/n a Bluetooth 4.1 LE. Díky těmto dvou možnostem se lze k Raspberry Pi 3 připojit i bezdrátově, což využívám i ve své práci pro pohodlnější manipulaci.

Rozložení GPIO pinů (Obr. 3), kterými se připojujeme k Rasberry Pi3 je stejné jako u předchozího modelu, takže je zajištěná kompatibilita s rozšiřujícími moduly. Disponuje 40 GPIO piny z toho 26 z nich je programovatelných a ostatní slouží k napájení nebo uzemnění.

Na raspberry Pi 3 existuje celá řada bezplatných operačních systému, jako je například Raspbian, Ubuntu mate, Risc OS, Windows 10 IOT core a další. Lze mít nainstalované i více operačních systémů. [5][6]

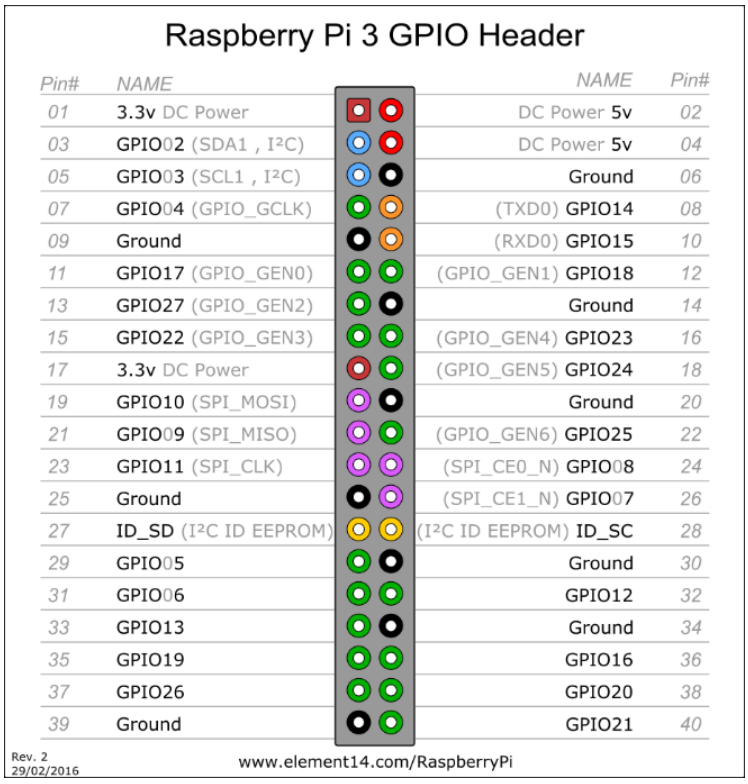


Obr. 2: Raspberry Pi3 Model B [7]

2.2.1 Specifikace

- SoC: Broadcom BCM2837 (CPU, GPU, DSP, SDRAM, jeden USB port)
- Procesor (CPU): 1.2GHz 64-bitový čtyřjádrový ARM Cortex-A53

- Video (GPU): Broadcom VideoCore IV @ 400 MHz / 300 MHz, OpenGL ES 2.0 (24 GFLOPS), MPEG-2 a VC-1 (s licencí), 1080p30 H.264/MPEG-4 AVC dekodér a kodér s vysokým profilem
- Paměť (SDRAM): 1 GB (sdílená s GPU)
- USB 2.0 porty: 4 (přes zabudovaný pětiportový USB hub, jeden USB port vnitřně propojen s ethernet portem)
- Video výstup: HDMI (rev 1.3 & 1.4), 14 HDMI rozlišení od 640×350 do 1920×1200 plus různé PAL a NTSC standardy, kompozitní video (PAL a NTSC) via 3.5 mm TRRS jack sdílený s výstupem zvuku
- Video vstup: 15-pinový MIPI konektor kamerového rozhraní (CSI)
- Zvukový výstup: Analog, HDMI, I2S
- Zvukový vstup: Prostřednictvím I2C rozhraní GPIO
- Integrovaná síť: 10/100 Mbit/s Ethernet + WiFi 802.11n a Bluetooth 4.1
- Nízkoúrovňové periferie: 17 x GPIO plus tytéž specifické funkce a HAT ID sběrnice
- Interní paměť: MicroSDHC slot
- Rozměry: 85,60 x 56,5 mm (bez konektorů)
- Hmotnost: 45 g



Obr. 3: Rozložení GPIO pinů [8]

2.3 Řídicí systém REX controls

*Řídicí systém REX (*Obr. 4*) je otevřený a škálovatelný systém vhodný pro vnořené řízení (embedded control), přenositelný na různé platformy s překladači jazyka C a C++ od jednoúčelových řídicích desek s jednoduchou exekutivou reálného času až po procesní stanice se standardními operačními systémy (Windows XP/Vista/7, Windows CE, Linux, PharLap ETS, apod.).*

Řídící systém REX je kompatibilní s programovým balíkem Matlab/Simulink. Tato kompatibilita je jednou ze základních myšlenek návrhu systému REX. Je možné využít veškeré možnosti Simulinku pro simulaci a odladění algoritmů. Po simulačním ověření lze řídicí algoritmy přeložit do binárních konfiguračních souborů, které je možné pomocí diagnostického protokolu založeného na standardu TCP/IP poslat přímo do cílových zařízení a podle nich zahájit řízení bez nutnosti odstavení zařízení. Ekvivalentní chování simulace a řízení v reálném čase zaručuje rozsáhlá knihovna funkčních bloků ve verzích jak pro Simulink tak i pro každou cílovou platformu.

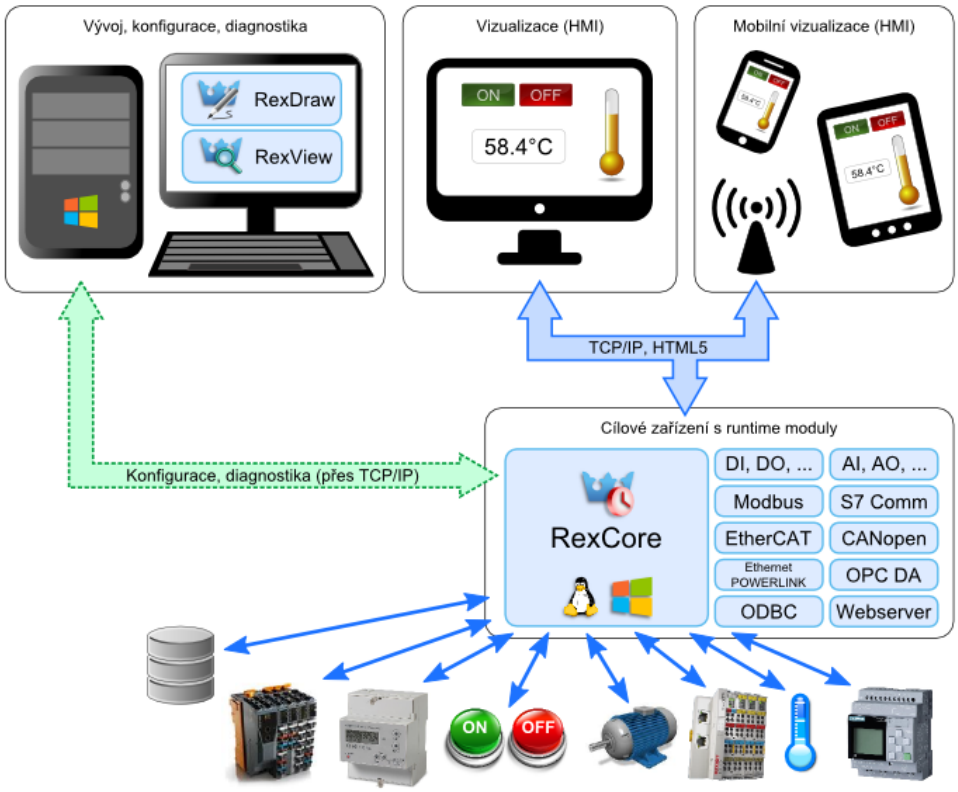
Na druhou stranu systém RE je plnohodnotným nástrojem i v případě, že uživatel nedisponuje licencí Simulink, jeho součástí je plnohodnotné programové vybavení pro vývoj a realizaci pokročilých řídicích systémů. [9]

Řídící systém REX je sada softwarových nástrojů pro tvorbu automatizovaných úloh, či matematicko-fyzikálních modelů. Řídící systém je možné uplatnit pro řízení strojů, robotů, měření a regulaci nebo ovládání výukových modelů. Uživatel může realizovat složité algoritmy, aniž by musel znát složitý programovací jazyk strukturovaného textu. Řídící systém REX obsahuje dvouhodinovou demoverzi, proto je možné jej vyzkoušet zcela zdarma. Po dvou hodinách se runtime jádro RexCore zastaví a je potřeba vytvořený program znovu nahrát. [9]

2.3.1 Základní součásti řídícího systému REX

- **RexDraw:** je grafické vývojové prostředí, kde se programují algoritmy pomocí funkčních bloků, které jsou součástí rozsáhlé knihovny samotného softwaru. K dispozici jsou funkční bloky pro generování signálů, logického řízení, matematické bloky, či bloky pro regulaci a zpracování analogových signálů. Vytvořené algoritmy se přeloží a nahrají skrze lokální síť nebo internet do cílového zařízení. Průběh probíhajícího algoritmu lze sledovat v reálném čase, tak jako všechny signály a stav funkčních bloků.
- **RexComp:** je překladač, který převádí vytvořený algoritmus v RexDraw, do binárního formátu. Při spuštění překladače informuje o překládaných souborech a případně informuje o vzniklých chybách.
- **RexView:** Je diagnostický nástroj, kterým lze sledovat průběh vytvořeného algoritmu řízeného jádrem řídicího systému REX. Připojení je možné v lokální síti i přes internet. Diagnostikuje problémy během provozu a informuje o všech výpočetních změnách algoritmu.

- **RexCore:** je nejdůležitější softwarový nástroj, který běží přímo na cílovém zařízení (Linux, IPC, WinPAC, Raspberry Pi, atd.). Zajišťuje spuštění a časování programu. Jednotlivé úlohy vykonává na základě nastavených priorit v režimu preemptivního multiaskingu.
- **RexHMI:** je nový nástroj přidaný v nejnovější verzi řídicího systému REX, který používá sadu programů a knihoven k tvorbě webové HMI. Jedná se o implementovaný a rozšířený open-source vektorový grafický editor Inkscape do softwaru, který slouží pro pohodlnější a plnohodnotnější vytvoření vizualizace našeho programu. Propojuje vytvořené grafické objekty s jednotlivými funkčními bloky, či parametry funkčního bloku a umožňuje sledovat a měnit hodnoty v reálném čase. Vizualizace je spustitelná na webových stránkách HTML5, které jsou podporovány moderními prohlížeči.

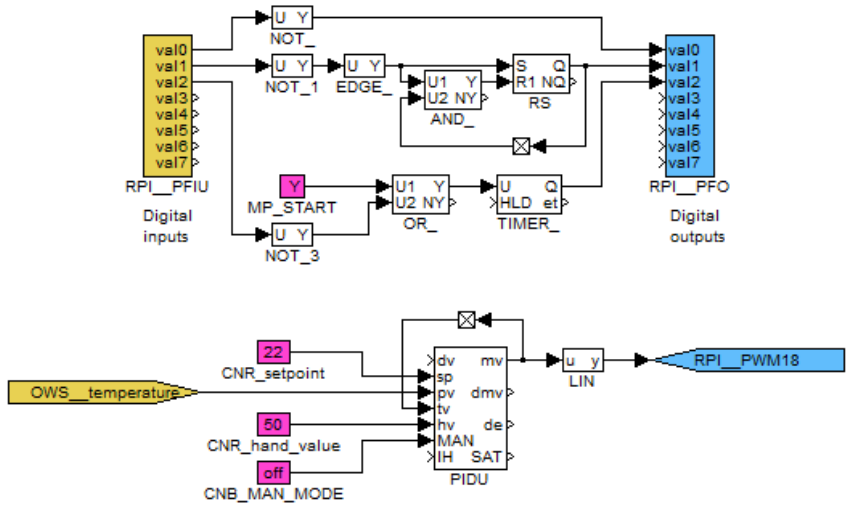


Obr. 4: Komponenty řídicího systému REX [10]

2.3.2 Programování v řídicím systému REX

Programování algoritmů řídicího systému REX probíhá na klasickém PC v grafickém vývojovém prostředí RexDraw. Algoritmy se programují funkčními bloky (Obr. 5) dle normy IEC 61131-3. Tato forma programování je známa při programování PLC automatů. Jazyk funkčních bloků dokáže nahradit mnoho řádků textového programu. Programování vzniká sestavováním a pospojováním jednotlivých funkčních bloků k sobě. Propojené funkce a funkční bloky vyjadřují

chování systému. Stejný způsob propojování je známý například i ze Simulinku, s kterým je řídicí systém REX kompatibilní. Řídicí systém REX obsahuje knihovnu funkčních bloků s časovači, komparátory, filtry, PID regulátory a mnoha dalšími.



Obr. 5: Přiklad algoritmů naprogramovaného funkčními bloky REXu [10]

2.3.3 Vlastnosti řídicího systému REX

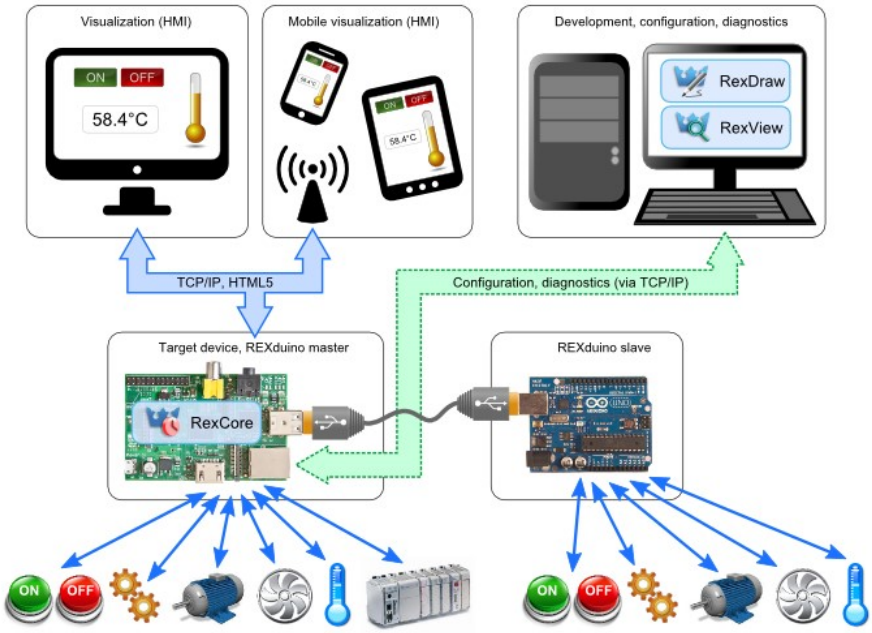
- Grafické programování bez ručního psaní kódu
- Programování řídicích jednotek na běžném PC nebo notebooku
- Uživatelské rozhraní pro desktop, tablet i smartphone
- Široká rodina podporovaných zařízení a vstupně-výstupních jednotek
- Algoritmy řízení prověřené průmyslovou praxí
- Snadná integrace do podnikových informačních systémů (ERP/BMS)

2.4 Platforma REXduino

Platforma REXduino (Obr. 6) vzniká spojením všech typů mikrokontroléru Arduina s počítačem Raspberry Pi. Kombinací těchto zařízení vzniká velmi levná a výkonná platforma pro řízení algoritmů v reálném čase. Komunikace mezi Raspberry Pi a Arduinem je postavena na softwaru REXduino. Symbióza řídicího systému REX a mikrokontrolérem Arduino, probíhá pomocí jednoduchého komunikačním protokolu, fungujícím na principu počítačového modelu master/slave. Do zařízení slave, které je v tomto případě Arduino, je nainstalován program, který zpracovává data přijímaná ze zařízení master přes USB připojení. Vykonává požadované operace a odpovídá zpět masteru. Master představuje Raspberry Pi. Zařízení slave funguje v podstatě jako

vstupní/výstupní jednotka. Na cílovém zařízení master neboli Raspberry Pi, je nainstalováno runtime jádro RexCore, kde běží hlavní část programu realizovaná funkčními bloky naprogramovanými na PC.

Všechny digitální i analogové GPIO piny jak u Raspberry Pi, tak u Arduina mohou být použity k interakci s okolním světem. Desky jsou navzájem propojeny USB kabelem. Tato kombinace využívá výhod výpočetního výkonu, paměti a ethernetového připojení ze strany Raspberry Pi s relativně bohatou množinou vstupů a výstupů na straně Arduina. [11]



Obr. 6: Principiální schéma platformy REXduino [11]

2.5 Vektorový grafický editor RexHMI designer

RexHMI designer je implementovaný vektorový grafický editor Inkscape do vývojového sowftwaru REX. Inskcape je open source vektorový grafický editor příbuzný svými funkcemi programům jako Sodipodi, Scribus, Illustrator, Freehand, CorelDraw nebo Xara X. Pro dokumenty používá formát SVG (Scalable Vector Graphics, škálovatelná vektorová grafika) za použití W3C standardu. Mezi podporované SVG schopnosti patří tvary, cesty, text, značky, klonování, průhlednost, změna velikosti, barevné přechody, vzorky a seskupování. Inkscape podporuje Creative Commons meta-data, k editaci vrstev, trasováním bitmap, textu na křivce, přímé editaci XML a dalšími funkcemi. Díky SVG jsou dostupné veškeré SVG filtry aplikovatelné na kresby k vytvoření většímu množství efektů. Inkscape je v současnosti v aktivním rozvoji a pravidelně jsou přidávány nové užitečné funkce. Obsahuje také desítky scriptů určených k ulehčení práce, generování objektů, změně jejich vlastností apod. Do Inkscape je možno importovat formáty jako JPEG, PNG, TIFF a další. Program běží v prvé řadě na GNU/Linux, nicméně je to multiplatformní

aplikace, která lze spustit i na Windows a Mac OS X. Hlavním cílem vývojářů Inkscape bylo vytvořit silný, schopný a volně dostupný nástroj na kreslení plně kompatibilní s XML, SVG a CSS standardy. Inkscape je vhodný pro tvorbu návrhů internetových stránek a při tvorbě grafiky pro web, proto je skvělým pomocníkem pro tvorbu vizualizace mé bakalářské práce v systému REX controls. Inkscape není určen primárně pro profesionální studia, ale i přes mnohé nedostatky se může rovnat v některých oblastech s dražšími komerčními programy.

Klady

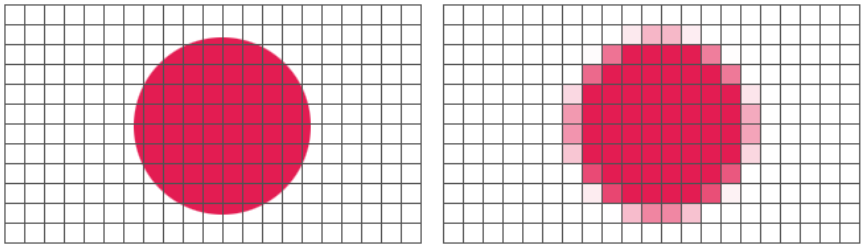
- Přívětivé a intuitivní ovládání pro začátečníky.
- Program je multiplatformní, pokud přejdete z Windows na Linux či Mac, přejde s vámi.
- Otevřený formát – lze přečíst i na konkurenčních programech.
- Velké množství funkcí.
- Cena – zdarma.

Zápory

- Neotevře některé formáty konkurence, hlavně CDR.
- Neobsahuje barvy PANTONE.
- Nespolupracuje s CMYK. Převod vytvořené grafiky do CMYK je nutno řešit v jiném programu. Nejlépe v prostředí Scribus.
- Stabilita. Občas nečekaně havaruje, ale obsahuje velice spolehlivou funkci obnovení.
- Při použití některých složitých funkcí zabere spoustu paměti.

2.5.1 Vektorová grafika

Pojmem vektorová grafika je označován jeden ze dvou hlavních způsobů zobrazení dvourozměrných obrázků. Vektorová grafika využívá k popisu obrázku přesně definovaných geometrických tvarů (body, přímky, mnohoúhelníky, křivky), čímž je popsán jakýkoliv tvar. Jednou z největších výhod vektorové grafiky je možnost jakékoliv změny velikosti obrázku, aniž by to mělo vliv na kvalitu obrázku. Druhým způsobem zobrazení je rastrová (bitmapová) grafika, kde je obrázek zobrazen pomocí hodnot jednotlivých barevných bodů (pixelů). Rozdíl mezi vektorovou a rastrovou grafikou je zobrazen na Obr. 7. [21]

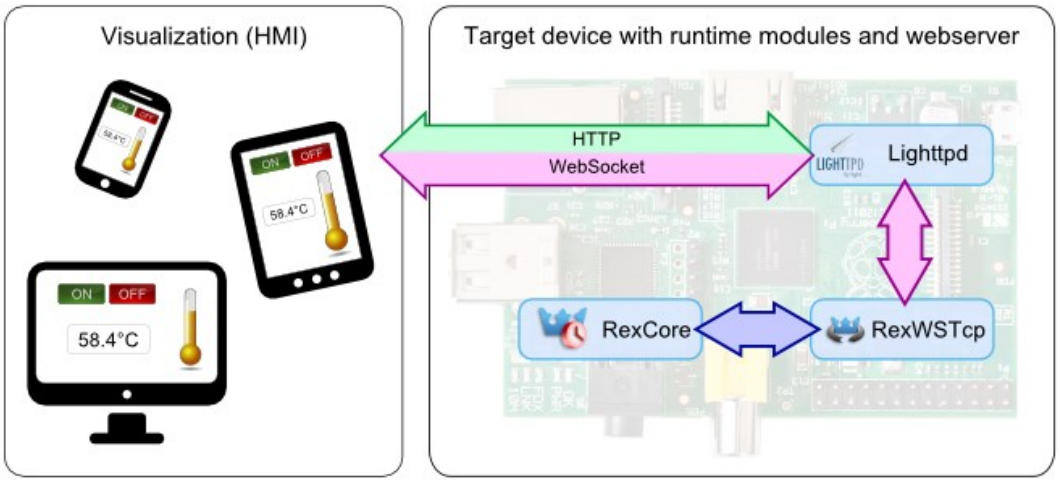


Obr. 7: Rozdíl mezi vektorovou a rastrovou grafikou [12]

RexHMI Designer obsahuje všechny potřebné nástroje a knihovny k vytváření HMI (Human–Machine Interface) tzn. rozhraní mezi člověkem a strojem. V podstatě se jedná o editor Inkscape s rozšiřujícími nástroji a knihovny pro vytváření vizualizace systému REX. Toto celé označujeme jako RexHMI. Vizualizaci můžeme vytvořit třemi různými zbůsoby, pomocí WebWatch, RexHMI Designer, WebBuDi.

- **WebWatch** automaticky generuje HMI ve vývojovém prostředí RexDraw při kompilování programu. Funkční blok HMI, který je součástí knihovny funkčních bloků, přesuneme do programové části s funkčním blokem *exec.mdl.* Tento blok se dá podobně parametrizovat jako jiné funkční bloky. Nastaví se parametr IncludeHMI a povolí se WebWatch vizualizace parametrem GenerateWebWatch. WebWatch je ideálním nástrojem pro okamžité vytvoření HMI, které je dostačující pro vývojáře soustav nebo podobných programů. Propojuje grafiku s téměř všemi signály řídícího algoritmu.
- **RexHMI Designer** vytváří klasický SVG soubor s příponou RexHMI. Je to skvělý nástroj pro vytváření grafického rozhraní HMI. Uživatelsky je přívětivý pro operátory a koncové uživatele. V podstatě se jedná o grafický editor Inkscape, s implementovanými funkcemi pro propojování grafických objektů se signály nebo parametry řídicího algoritmu.
- **WebBuDi** je zkratka tří slov *web*, *buttons* and *displays*. V překladu to znamená web, tlačítka a displeje. Jedná se o vytváření skriptů v JavaScriptu s deklarovanými bloky, které popisují propojení datových bodů s řídícím algoritmem. Poskytuje interakci textových objektů s vybranými signály řídícího algoritmu. Vhodný je především pro vývojáře složitějších systémů, nebo může soužit k editaci v nouzovém režimu HMI pro nestandartní situace.

Výsledkem každého z výše uvedených nástrojů, je vytvoření webové stránky HTML5 (podporováno všemi moderními webovými prohlížeči), která je generovaná z interního webového serveru REX. K serveru je možné přistupovat z počítače, tabletu a také mobilního zařízení. HMI je přístupné z interního serveru cílového zařízení a je zobrazeno webovým prohlížečem (doporučovány jsou Google Chrome a Mozila Firefox). Distribuovány jsou vlastním webserverem přes standartní http protokol. Po načtení stránky v prohlížeči se vytvoří websocketové spojení mezi prohlížečem a jádrem RexCore, běžícím na Raspberry Pi. Webový server Lighttpd a websocketový server RexWSTcp pracují jako prostředníci mezi uživatelem a řídícím algoritmem. Tím lze snadno nastavovat a sledovat signály nebo parametry běžícího programu. Struktura RexHMI je popsána na Obr. 8. [13]



Obr. 8: Struktura RexHMI [14]

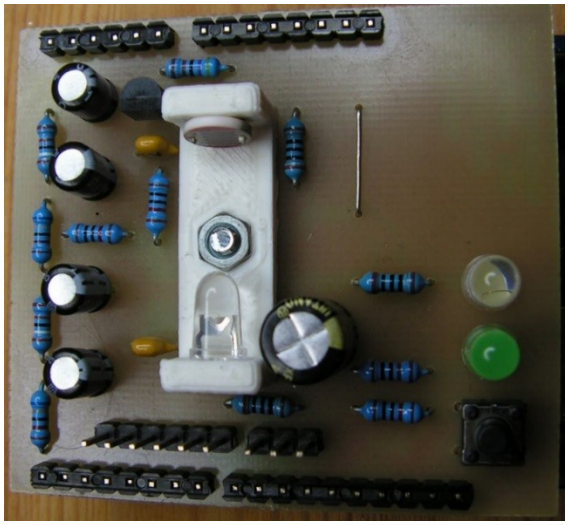
2.6 Testovací stojan s výukovým modelem

Jako výukový model byl navrhnut a realizován plošný spoj (viz. Obr. 9), simulující jednoduché fyzikální úlohy pro praktické znázornění regulačních obvodů do předmětu Kybernetika. Plošný spoj se skládá z několika elektronických obvodů složených z následujících součástek:

- 2x rezistor 100 kΩ
- 2x rezistor 390 kΩ
- 3x rezistor 10 kΩ
- 1x rezistor 1 kΩ
- 2x rezistor 100 Ω
- 1x fotorezistor GES – GES05101428
- 4x elektrolyticky kondenzátor 47 µF/10 V
- 1x elektrolyticky kondenzátor LOW-ESR 220 µF/16 V
- 2x keramický kondenzátor 100 ηF
- 1x operační zesilovač TS912ID
- 1x teplotní čidlo DALAS DS18B20
- 2x LED cool white 5 mm 518-091
- 1 LED green 5 mm
- 1x Mikrospínač TC-0103-T
- 1x pinové kolíky 832-021

Elektronické obvody lze přes řídící systém REX sledovat a regulovat. Prvky na plošném spoji je díky REX controls možné programově mezi sebou propojit tak, aby bylo možné nasimulovat jednotlivé fyzikální úlohy. Pro plošný spoj jsou připraveny následující úlohy:

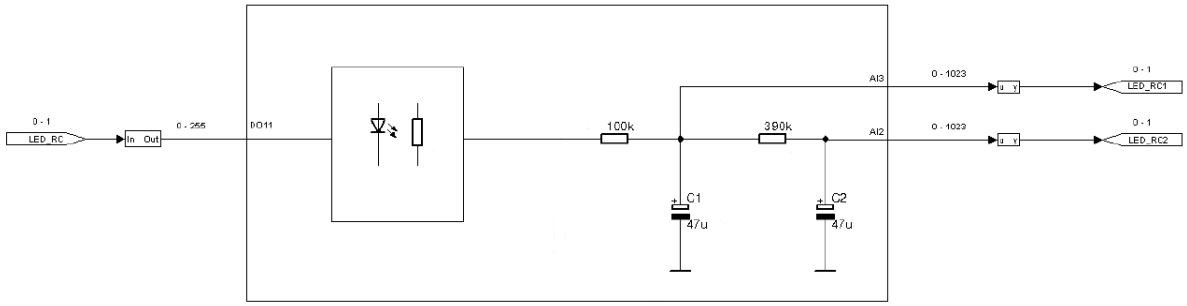
- LED dioda ovládaná tlačítkem.
- LED dioda řízená pulzně šířkovou modulací.
- LED dioda řízená proměnnou.
- Soustavu 2. řádu složenou dvěma RC články v sérii.
- Soustavu 2. řádu řízená LED diodou přes fotorezistor.
- Řízení servomotoru v oblasti od -45 ° do 45 °.



Obr. 9: Výukový model

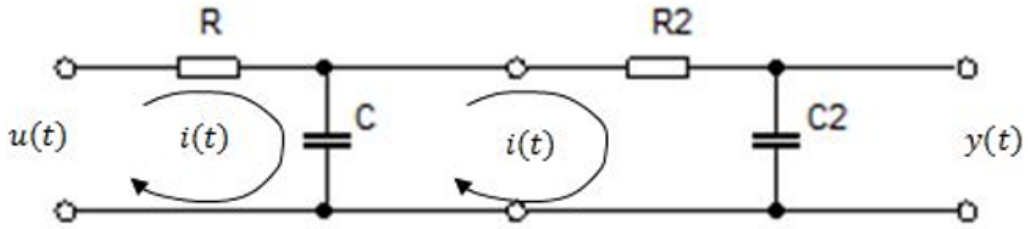
2.6.1 Soustava 2. řádu řízená LED diodou přes fotorezistor

Pro moji závěrečnou práci jsem se zaměřil na soustavu 2. řádu řízenou LED diodou přes fotorezistor (Obr. 10). Tato soustava je zapojena poté zapojena do zpětnovazebního regulačního obvodu realizovaném v řídícím systému REX. Regulovanou veličinou je svit LED diody, který je emitován na regulovanou soustavu (fotorezistor spojený se vstupem dvou sériově spojených RC článků, simulujících soustavu 2. řádu). Na emitované světlo z LED diody dopadající na fotorezistor působí vnější prostředí a tím simulujeme chyby v regulačním obvodu. Čím větší je požadována hodnota výstupního signálu, tím větší světlo emituje LED dioda na fotorezistor a naopak.



Obr. 10: Elektrotechnické schéma fyzikální úlohy

Pokud by se jednalo o zapojení dvou RC články v sérii s galvanickým oddělením, mohli bychom na tento systém pohlížet z pohledu teorie systémů jako na zapojení dvou přenosů v sérii, po jejichž vynásobení získáme celkový přenos. Ale jelikož vstup druhého článku impedančně zatěžuje první článek, nelze na tento systém pohlížet z pohledu teorie systémů. Je nutné na něj nahlížet jako na jedno zapojení, které je třeba analyzovat jako celek (Obr. 11).



Obr. 11: Elektrické schéma zapojení dvou RC článku v sérii

Za použití metody smyčkových proudů, která využívá II. Kirchhoffova zákona, zjistíme všechny neznáme v obvodu (úbytky napětí a proudy na jednotlivých prvcích). Poté vypočteme celkový přenos. První krok pro řešení obvodu pomocí metody smyčkových proudů je označení nezávislých smyček $i_{(t)}$ (Obr. 11). Poté šipkou označíme jejich orientaci, kterou si volíme libovolně, ale v následujících krocích si ji musíme hlídat. Nyní můžeme začít skládat rovnice. Pro první smyčku vzniká rovnice (1) a pro smyčku druhou rovnice (2).

$$R_1 * I_1 + \frac{1}{C_1 s} * (I_1 - I_2) = \mathrm{U} \tag{1}$$

$$\frac{1}{C_1 s} * (I_1 - I_2) + R_2 * I_2 + \frac{1}{C_2 s} * I_2 = 0 \tag{2}$$

Po drobné matematické úpravě osamostatníme I_1 a I_2. a získáme upravený tvar rovnice (1) a rovnice (2) na rovnici (3) a rovnici (4).

$$I_1 * \left(R_1 + \frac{1}{C_1 s}\right) + I_2 * \left(-\frac{1}{C_1 s}\right) = \mathrm{U} \tag{3}$$

$$\left(-\frac{1}{C_1 s}\right) + I_2 * \left(\frac{1}{C_1 s} + R_2 + \frac{1}{C_2 s}\right) = 0 \tag{4}$$

Nyní můžeme upravené rovnice dosadit do maticového tvaru (5), ze kterého si odvodíme celkový přenos dvou RC článku zapojených do série (7). Ten je popsán v Laplacově transformaci ve tvaru racionální lomené funkce. [15]

$$\begin{bmatrix} R_1 + \frac{1}{C_1 s} & -\frac{1}{C_1 s} \\ -\frac{1}{C_1 s} & \frac{1}{C_1 s} + R_2 + \frac{1}{C_2 s} \end{bmatrix} * \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} U \\ 0 \end{bmatrix} \tag{5}$$

$$I_2 \rightarrow U_2 \rightarrow G = \frac{U_2}{U_1} = \frac{1}{R_1 C_1 R_2 C_2 s^2 + R_1 C_1 s + R_2 C_2 s + R_1 C_2 s + 1} \tag{6}$$

$$G = \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_1 C_2) s + 1} \tag{7}$$

$$G = \frac{1}{86{,}15 s^2 + 27{,}73 s + 1} \tag{8}$$

$$T_1 = 24{,}16\ s \text{ a } T_2 = 3{,}57\ s \tag{9}$$

2.7 Identifikace soustavy

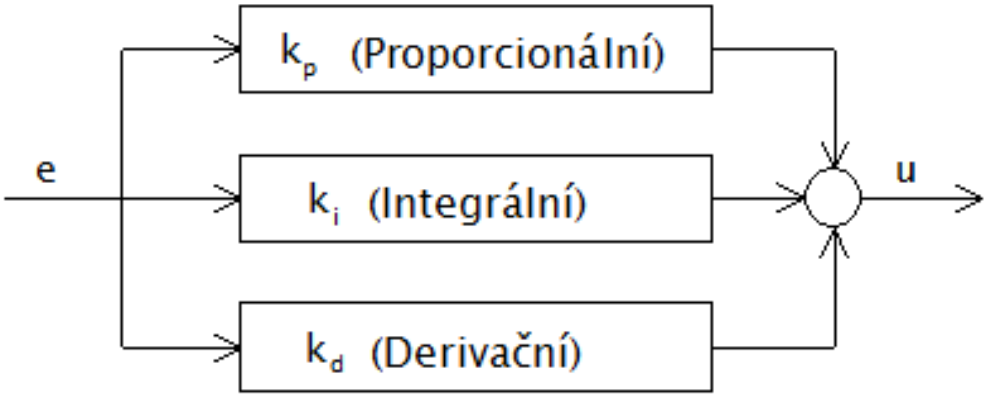
Identifikací soustavy získáme matematicko-fyzikální model řízeného systému. K tomuto popisu řízeného systému lze dojít analyticky nebo experimentálně.

U analytické metody je třeba znát matematické a fyzikální zákony, které jsou aplikovány na soustavu, získá se tím soustava několika rovnic popisující charakter systému. Metoda vyžaduje teoretické a praktické znalosti o identifikovaném systému. Analytickou identifikaci jsem provedl v předchozí kapitole (kap. 2.6) rovnice (8), kde časové konstanty vyšli $T_1 = 24.16$ s a $T_2 = 3{,}57$ s.

Experimentální metodou se matematický model získá z naměřených výstupních dat řízené soustavy. Vybudíme systém vhodně zvoleným signálem (jednotkovým skokem) přivedeným na vstup soustavy a na výstupu sledujeme chování systém. Získaná data se analyzují, například pomocí Identification toolbox, který je určen pro vytváření matematických modelů z naměřených dat v programu Matlab.

Experimentální metodu jsem zvolil pro moji práci, kdy jsem jednotkovým skokem vybudil soustavu a přes řídicí systém REX jsem naměřil výstupní data, která jsem identifikoval nástrojem Ident. Touto problematikou se zabývám v praktické části.

2.8 PID regulátor



Obr. 12: Blokové schéma paralelního zapojení PID regulátoru

Proporcionálně integračně derivační regulátor (Obr. 12) je složený regulátor, který patří mezi spojité regulátory a spolu s PI regulátorem patří v praxi mezi nejvíce používané. Je složený z proporcionální, integrační a derivační části. Proporcionální část funguje jako prostý zesilovač (zvyšuje regulační odchylku), takže akční veličina na výstupu regulátoru je přímo úměrná regulační odchylce na vstupu regulátoru. U integrační části je akční veličina integrací regulační odchylku, takže i při nulovém vstupním signálu je výstup nenulový (určitá změna na vstupu odpovídá určité poloze na výstupu). Integrační část se stará o odstranění trvalé odchylky. Derivační část je v praxi samostatně nerealizovatelná, takže se používá v kombinaci s regulátorem PD nebo PID. Výstup regulátoru je přímo úměrný derivaci regulační odchylky (určitá změna rychlosti na vstupu odpovídá určité poloze regulačního orgánu). Derivační část v předstihu kompenzuje změny regulované veličiny, proto se používá k tlumení zákmitu regulačního pochodu. Celkovou akční veličinu PID regulátoru můžeme vyjádřit jako součet P-regulátoru, I-regulátoru a D-regulátoru. Obecný tvar paralelního tvaru je vyjádřen rovnicí (10). [16]

$$u(t) = \mathrm{r}_0 * e(t) + r_d \frac{de_{(t)}}{dt} + ri * \int_0^t e(\tau)d\tau + u(0) \qquad (10)$$

Je třeba si uvědomit, že základem správné funkce regulačního obvodu jsou optimálně nastavené parametry regulátoru. To závisí na míře znalosti regulované soustavy, proto využíváme identifikaci systému, ze které získáme potřebné informace pro navržení PID regulátoru.

2.9 Návrh spojitého PID regulátoru

Pro nastavení PID regulátoru existuje mnoho metod. Rozdělujeme je na analytické, experimentální a kombinované, tj. analyticko-experimentální. Všechny mají jedno společné, nalezení vhodného regulátoru a zajištění kvality regulačního pochodu. Analytické metody vychází ze znalosti matematicko-fyzikálního modelu řízeného systému a předpokládaného modelu regulátoru. Experimentální metody jsou založeny na vyhodnocování z naměřených dat řízeného procesu. Parametry se pak získají výpočty právě z těchto naměřených dat, podle definovaných vztahů.

Známé metody jsou například Ziegler-Nicholsova metoda, metoda kritických parametrů, metoda skokové odpovědi, metoda optimálního modulu, metoda požadovaného modelu, metoda minimalizace podle kritéria ITAE, atd. Pouze některé z nich, se ale dají použít na zkoumanou soustavu. Zaměřil jsem se proto jen na ukázku dvou z nich, které poté budu aplikovat na mou soustavu.

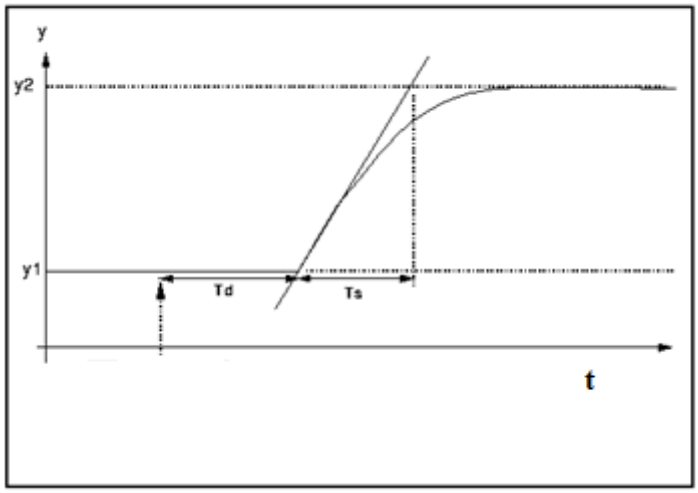
2.9.1 Ziegler-Nicholsova metoda

Je to jedna z nejstarších metod pro nastavení regulátoru a jedná se o metodu experimentální. Publikována byla již v roce 1942 a výsledek byl prokázán i teoreticky. Metoda má charakter velmi jednoduchého algoritmu a v praxi se proto používá pro první odhad parametrů regulátoru. V podstatě tato metoda má dvě varianty aplikování. Pro aplikaci této metody je třeba počítat s převodem PID regulátoru z paralelního tvaru na PID regulátor v sériovém tvaru (11).

$$G_r = k * \left(1 + \frac{1}{Ti * s} + Td * s\right) \tag{11}$$

1. **Varianta – pro otevřené regulační obvody**

Na vstup regulovaného procesu přivedeme skokový signál a zaznamenáme přechodovou charakteristiku. Z přechodové charakteristiky (Obr. 13) určíme parametry, kterými se přechodová charakteristika prezentuje: *Td* - doba průtahu (dopravní zpoždění), *Ts* - doba náběhu (časová konstanta) a *k* - statické zesílení. Statické zesílení se získá podle rovnice (12). [17]



Obr. 13: Přechodová charakteristika pro ladění obvodu s otevřenou smyčkou

Posledním krokem je vypočtení parametrů PI nebo PID regulátoru dle Tab. 1, kde jsou uvedeny vztahy pro určení jednotlivých parametrů

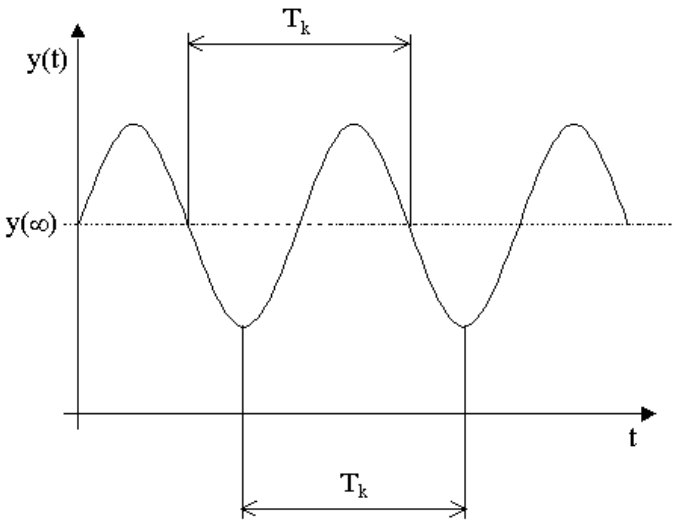
Tab. 1: Vztahy pro výpočet parametrů regulátoru s otevřenou smyčkou

Regulátor	K	T_i	T_d
PI	0,9*k	3,33*T_d	
PID	1,2*k	2*T_d	0,5*T_d

$$k = \frac{\Delta y}{\Delta u} \tag{12}$$

2. Varianta – pro uzavřené regulační obvody

U této metody přivedeme regulační obvod do tzv. kritického stavu, v tomto stavu začíná se objeví netlumené kmity s konstantní amplitudou (Obr. 14), ale regulátor je stále v mezi stability. Pokud kritického stavu nedosáhne, neleze tuto metodu použít. Regulátor pracuje pouze s proporcionální složkou, integrační a derivační složka je tedy vyřazena. Do tohoto stavuje jej přivedeme postupným zvyšováním zesílením regulátoru.



Obr. 14 : Určení kritické periody

Tímto postupem získáme dvě hodnoty, hodnotu zesílení regulátoru k_p a periodou kritických kmitů T_k. Tak jako u první varianty, i zde z těchto získaných hodnot vypočteme parametry regulátoru dle vztahů z uvedených v Tab. 2. [18]

Tab. 2: Vztahy pro výpočet parametrů regulátoru s uzavřenou smyčkou

Regulátor	K	T_i	T_d
PI	0,45*k	0,85*T_k	
PID	0,6*k	0,5*T_k	0,125*T_k

2.9.2 Metoda požadovaného modelu (dříve metoda inverzní dynamiky)

Metoda se používá pro snadné a rychlé navržení analogových, ale i číslicových regulátorů pro základní druhy regulovaných soustav. Umožňuje seřizování jak číslicových regulátorů pro diskrétní regulační obvody, tak i spojitých regulátorů pro spojité regulační obvody. Pro použití této metody se klade důraz na tvar přenosu regulované soustavy, přenos musí být uveden v jednom ze základních tvarů. Očekává se, že přenos soustavy bude s dopravním zpožděním. Jedná se analyticko-experimentální metodu. Regulace vhodné pro použití této metody shrnuje Tab. 3.

Tab. 3: Základní tvary přenosové soustavy

Regulovaná soustava		Regulátor < Analogový $T=0$ Číslicový $T>0$				
		Typ	k_p^*		T_I^*	T_D^*
			$T_d = 0$	$T_d > 0$		
1	$\frac{k_1}{s} \cdot e^{-T_d s}$	P	$\frac{2}{k_1(2T_w + T)}$	$\frac{a}{k_1}$	---	---
2	$\frac{k_1}{T_1 s + 1} \cdot e^{-T_d s}$	PI	$\frac{2T_I^*}{k_1(2T_w + T)}$	$\frac{aT_I^*}{k_1}$	$T_1 - \frac{T}{2}$	---
3	$\frac{k_1}{s(T_1 s + 1)} \cdot e^{-T_d s}$	PD	$\frac{2}{k_1(2T_w + T)}$	$\frac{a}{k_1}$	---	$T_1 - \frac{T}{2}$
4	$\frac{k_1}{(T_1 s + 1)(T_2 s + 1)} \cdot e^{-T_d}$ $T_1 \geq T_2$	PID	$\frac{2T_I^*}{k_1(2T_w + T)}$	$\frac{aT_I^*}{k_1}$	$T_1 + T_2 - T$	$\frac{T_1 T_2}{T_1 + T_2} - \frac{T}{4}$
5	$\frac{k_1}{T_0^2 s^2 + 2\xi_0 T_0 s + 1} \cdot e^{-T_d s}$ $0{,}5 < \xi_0 \leq 1$	PID	$\frac{2T_I^*}{k_1(2T_w + T)}$	$\frac{aT_I^*}{k_1}$	$2\xi_0 T_0 - T$	$\frac{T_0}{2\xi_0} - \frac{T}{4}$

Tab. 4: Závislost koeficientu α a β na koeficientu k

k	0	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
α	1,282	0,984	0,884	0,761	0,697	0,669	0,669	0,64	0,618	0,599	0,577
β	2,718	1,944	1,72	1,561	1,437	1,337	1,248	1,172	1,104	1,045	0,992

2.9.3 Metoda minimalizace kritéria ITAE

Metoda minimalizace podle kritéria (Integral time and error) je založena na hledání minima umíněného kritéria, jehož vyjádření je uvedeno rovnicí. (13)

$$I_{ITAE} = \int_0^\infty |e_{(t)}| * t * dt \tag{13}$$

Najít řešení je často velmi komplikované, v některých případech dokonce nemožné. K řešení je třeba použít numerických metod a výpočetní techniky. Z přenosu uzavřené smyčky se určí přenos chyby $E_{(s)}$ a zpětnou Laplaceovou transformací se vypočítá funkce chyby $e_{(t)}$. Na závěr se hledá minimum této funkce. Přenos PID regulátoru je daný rovnicí. (14)

$$G_r(\mathrm{s}) = \mathrm{r}_0 + \frac{r_i}{s} + r_d s * s \tag{14}$$

Použitím následujících numerických vztahů získáme proměnné r_0, r_i a r_d. (15)(16)(17)

$$K_r = \mathrm{r}_i = \frac{1}{\mathrm{T}_i} \tag{15}$$

$$T_r = \frac{\mathrm{r}_0}{\mathrm{r}_i} \tag{16}$$

$$\mathrm{r}_0 = \frac{T_r}{\mathrm{T}_i} \tag{17}$$

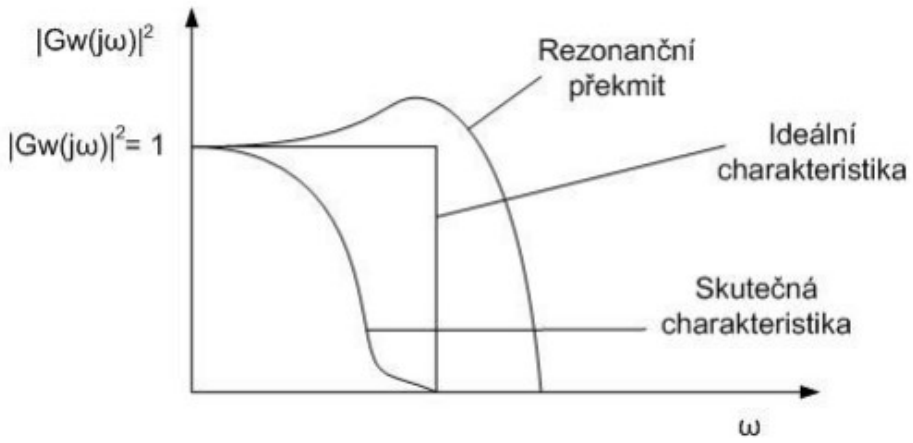
Přenos otevřeného regulátoru má tvar (18) a nakonec přepočteme pro přenos uzavřenou smyčkou. (19)

$$G_o(\mathrm{s}) = G_r(\mathrm{s}) * G(\mathrm{s}) \tag{18}$$

$$G_w(\mathrm{s}) = \frac{G_0(\mathrm{s})}{1 + G_0(\mathrm{s})} \tag{19}$$

2.9.4 Metoda optimálního modulu

Metoda vychází z frekvenční vlastnosti uzavřeného zpětnovazebního obvodu (Obr. 15). Požadavkem je, aby byl modul přenosu uzavřené smyčky roven přibližně 1 a průběh amplitudové frekvenční charakteristiky uzavřené smyčky byl monotónně klesající bez rezonančních překmitů.



Obr. 15: Grafické znázornění požadavků na metodu optimálního modulu

Budeme-li uvažovat, že přenos je dle rovnice (20), bude také platit

$$G_w(\mathrm{s}) = \frac{b_m s^m + \cdots + b_1 s^1 + b_0}{a_n s^n + \cdots + a_1 s^1 + a_0} \tag{20}$$

a

$$A_w^2(\mathrm{j}\omega) = G_w(\mathrm{j}\omega) G_w(\mathrm{j}\omega) = \frac{B_m \omega^{2m} + \cdots + B_1 \omega^2 + B_0}{A_n \omega^{2n} + \cdots + A_1 \omega^2 + A_0} \tag{21}$$

Nyní určíme koeficienty jmenovatele a čitatele. [19]

$$A_0 = a_0^2 \tag{22}$$

$$A_1 = a_1^2 - 2a_0 a_2 \tag{23}$$

$$A_2 = a_2^2 - 2a_1 a_3 + 2a_0 a_4 \tag{24}$$

$$A_2 = a_2^2 - 2a_1 a_3 + 2a_0 a_4 \tag{25}$$

$$A_i = a_i^2 + 2 \sum_{j=1}^{i} (-1)^j \, a_{i-j} a_{i+j} \tag{26}$$

$$B_0 = b_0^2 \tag{27}$$

$$B_1 = b_1^2 - 2b_0 b_2 \tag{28}$$

$$B_2 = b_2^2 - 2b_1 b_3 + 2b_0 b_4 \tag{29}$$

$$B_i = b_i^2 + 2 \sum_{j=1}^{i} (-1)^j \, b_{i-j} b_{i+j} \tag{30}$$

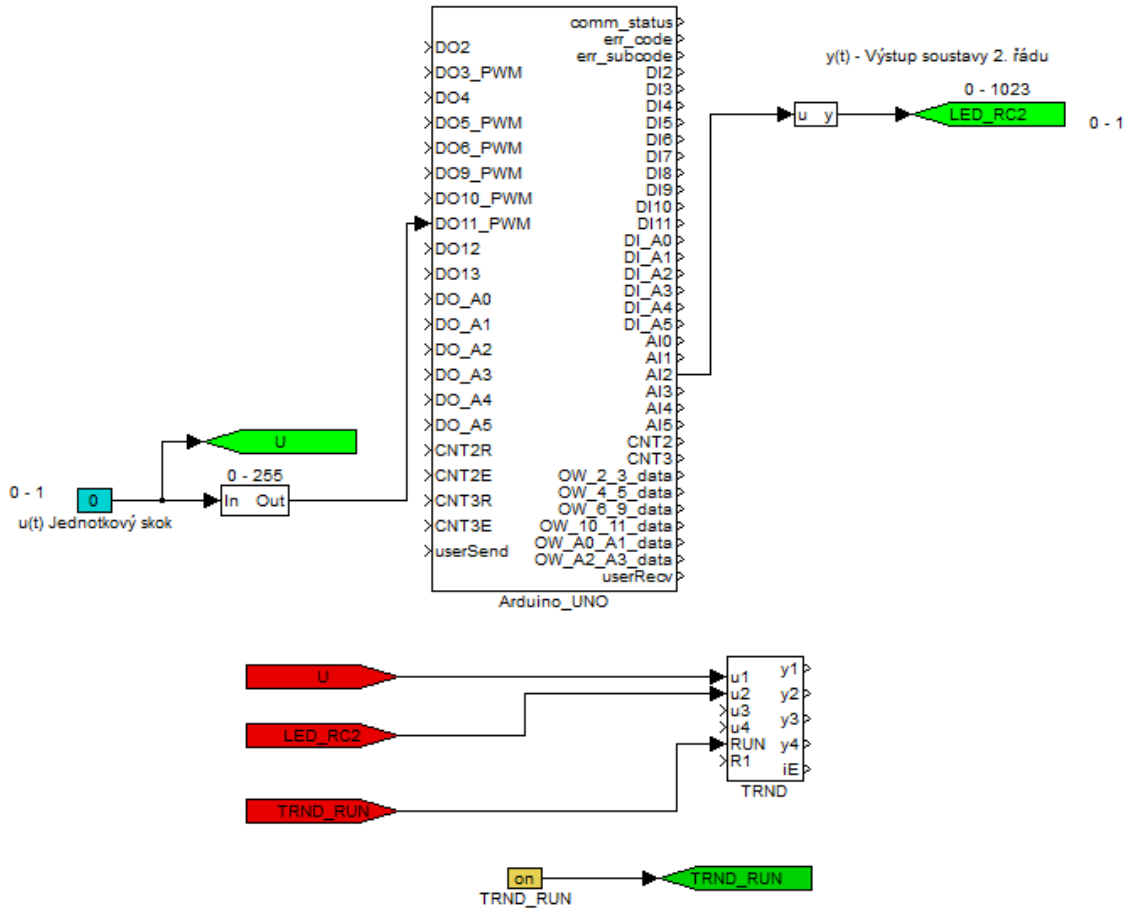
Pro vyhovující průběh regulačního průběhu je dostačující, aby byla splněna podmínka. (31)

$$\frac{B_0}{A_0} \geq \frac{\mathrm{B}_i}{\mathrm{A}_i} \tag{31}$$

3 Praktická část

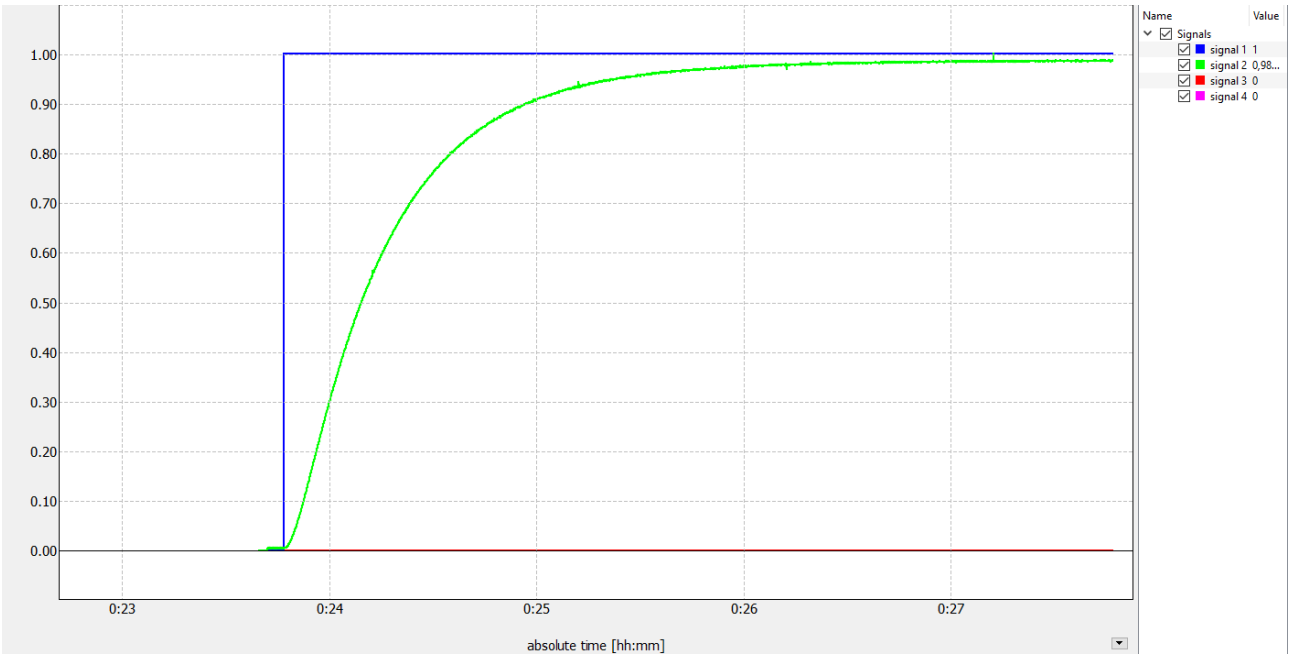
3.1 Identifikace zvolené soustavy

Pro identifikaci systému bylo třeba zjistit přechodovou charakteristiku, kterou jsem získal přivedením jednotkového skoku na vstup regulované soustavy a změřením odezvy na výstupu soustavy vytvořeném modelu řídícího algoritmu (Obr. 16). Tyto data jsem exportoval do programu Matlab, kde za pomoci Identification Toolbox, jsem provedl identifikaci soustavy. Před samotnou identifikací je nutno upravit data tak, aby neobsahovala neznáme stavy před samotným přechodovým dějem.



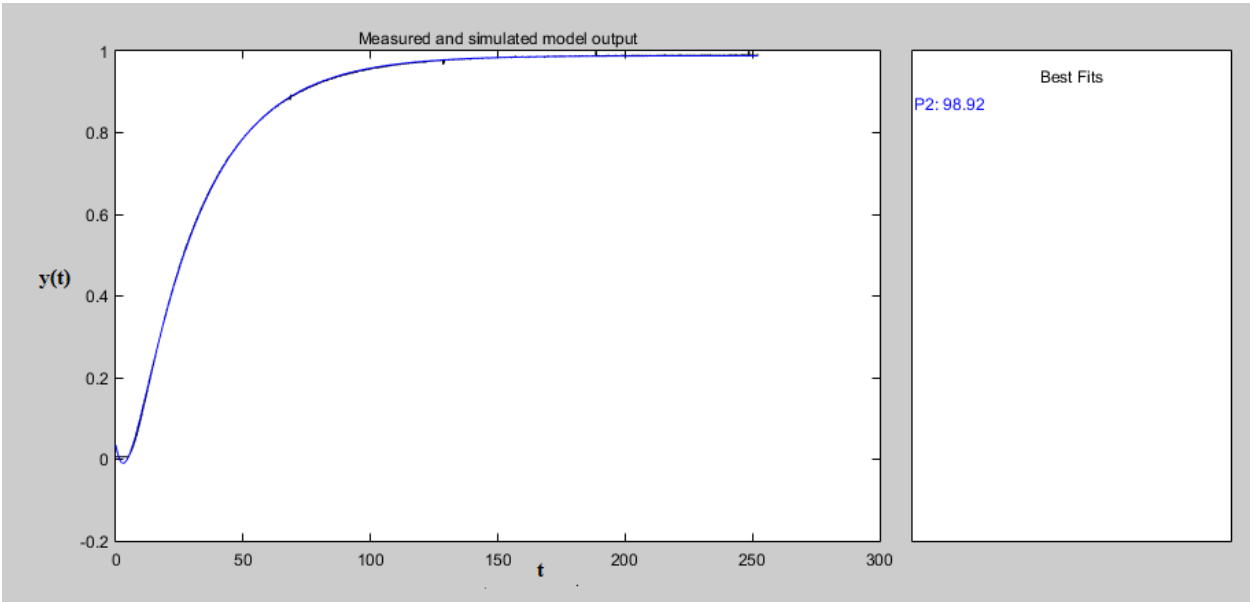
Obr. 16: Model simulující jednotkový skok

Ve funkčním bloku jednotkový skok jsem nastavil hodnotu 1 a blokem TRND jsem zaznamenal naměřená data. Po ustálení přechodového děje jsem zaznamenaná dat exportoval do Excelu, a dále je importoval do pracovního prostoru Workspace v Matlabu. Přechodový děj vyvolaný jednotkovým skokem je znázorněn v grafu (Obr. 17).



Obr. 17: Odezva jednotkového skoku naměřená v řídícím systému REX

Pro provedení identifikace v Identification Toolbox jsem nastavil parametry pro vyhledání sytému druhého řádu. Identifikace proběhla v pořádku (Obr. 18).



Obr. 18: Porovnání naměřených dat s odezvou systému druhého řádu

Identifikovaný přenos druhého řádu odpovídá na 98,92 % a výsledný přenos soustavy druhého řádu je (32).

$$G_{(s)}\frac{0{,}97895}{(25{,}671s+1)(4{,}8657s+1)} \tag{32}$$

3.2 Návrh PID regulátoru

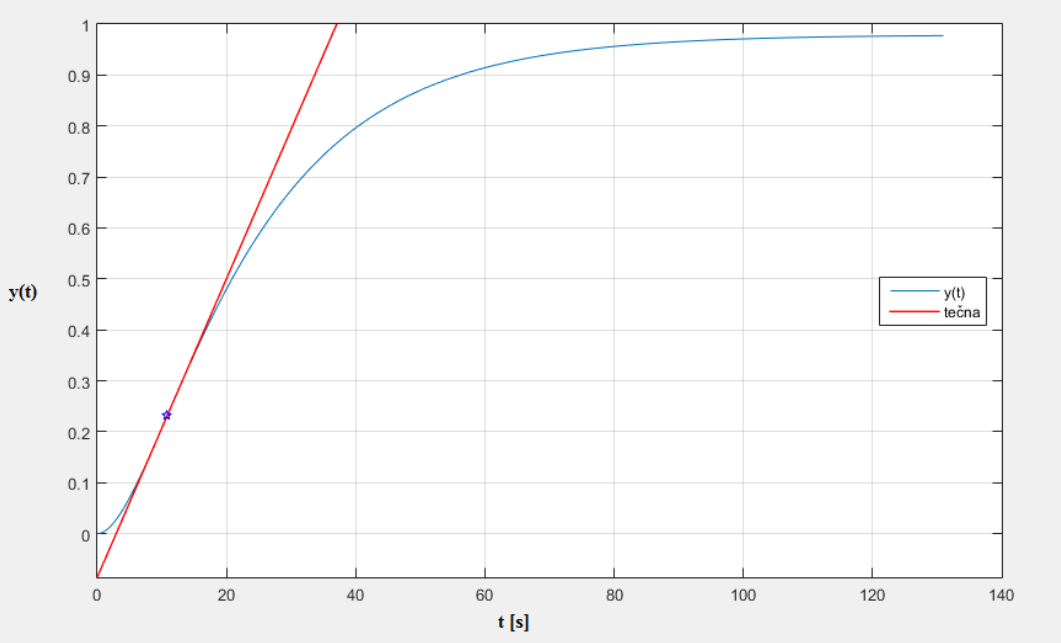
K návrhu regulátoru jsem použil metody

- **Ziegler-Nicholsova**
- **Metodu optimálního modulu**

3.2.1 Ziegler-Nicholsova metoda

Ziegler-nicholsovu metodu jsem vybral pro její jednoduchost a jednoznačnost od prvního kroku. Podle přechodové charakteristiky je vidět, že se jedná o přetlumený statický systém druhého řádu, který je statický a nelze rozkmitat. Proto použiji její modifikovanou verzi, která pracuje s odečtenými parametry z přechodové charakteristiky. Koeficienty regulátoru získáme výpočtem ze vztahů dle Tab. 1. Nepracujeme tedy s kritickými hodnotami obvodu ani s periodou kmitů.

Dobu průtahu T_u a dobu náběhu T_n lze určit ze směrnice tečny k inflexnímu bodu. Za pomoci Matlabu jsem vytvořil program, pro vykreslení přenosu s tečnou v inflexním bodu (Obr. 19).



Obr. 19: Určení tečny pro metodu ZN

Program vypočte i dobu průtahu T_u dobu náběhu T_n a zesílení k.

$$T_u = 2{,}9192 \tag{33}$$

$$T_n = 33{,}3654 \tag{34}$$

$$k = 0{,}9774 \tag{35}$$

Tvar PI regulátoru program vypočítal následně (36).

$$G_{r_{PI}} = k_p * \left(1 + \frac{1}{T_i \cdot s}\right) \tag{36}$$

$$k_p = 0{,}9 * \frac{T_n}{k_1 \cdot T_u} = 10{,}5250 \tag{37}$$

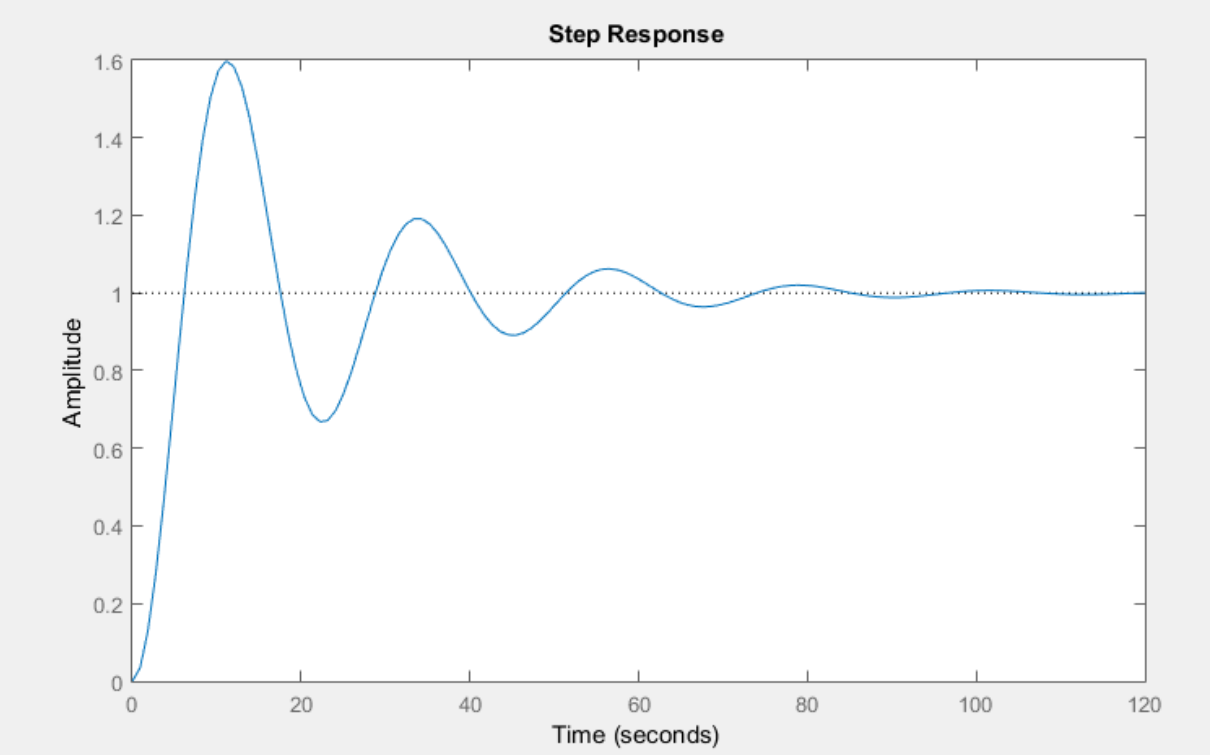
$$T_i = 3{,}33 * T_u = 9{,}7209 \tag{38}$$

$$T_d = 0{,}5 * T_u = 1{,}4596 \tag{39}$$

Výsledný přenos regulátor PI je pak (40).

$$G_r = \frac{102{,}3s + 10{,}53}{9{,}721s} \tag{40}$$

Z průběhu (Obr. 20) je viditelné, že překmit regulátoru je poměrně vysoký, což není pro tuto metodu neobvyklé. Jedná se o experimentální metodu, proto je překmit vysoký 59,8 % a přenos bude ustálený za 79,5 s.



Obr. 20: Regulace PI regulátorem metodou ZN

3.2.2 Metoda optimálního modulu

Řešení bylo provedeno na půl Matlabem a na půl jsem jej zpracovával ručně. To je u této metody nevýhodou, protože může dojít k přehlédnutí a následně chybě výpočtu. Nejprve je třeba vypočítat a zjednodušit přenos v Matlabu. Z tohoto přenosu pak je třeba ručně určit b_1, b_0, a_3, a_2, a_1, a_0 (41).

$$G_r = \left(\frac{9789500\,\mathrm{T\,ks} + 9789500}{10000000\,\mathrm{T\,s} + 305367000\,\mathrm{T\,s} + 1249073847\,\mathrm{T\,s} + 9789500\,\mathrm{T\,k\,s} + 9789500}\right) \quad (41)$$

$$b_0 = 9789500 \quad (42)$$

$$b_1 = 9789500 * T * ks * \quad (43)$$

$$a_0 = 9789500 \quad (44)$$

$$a_1 = 10000000 * T + 9789500 * T * k \quad (45)$$

$$a_2 = 305367000 * T \quad (46)$$

$$a_3 = 1249073847 * T \quad (47)$$

A následně určíme koeficienty optimálního modulu

$$B_0 = {b_0}^2 \tag{48}$$

$$B_1 = {b_1}^2 - 2 * b_0 * b_2 \tag{49}$$

$$B_2 = {b_2}^2 - 2 * b_1 * b_3 + 2 * b_0 * b_4 \tag{50}$$

$$A_0 = {a_0}^2 \tag{51}$$

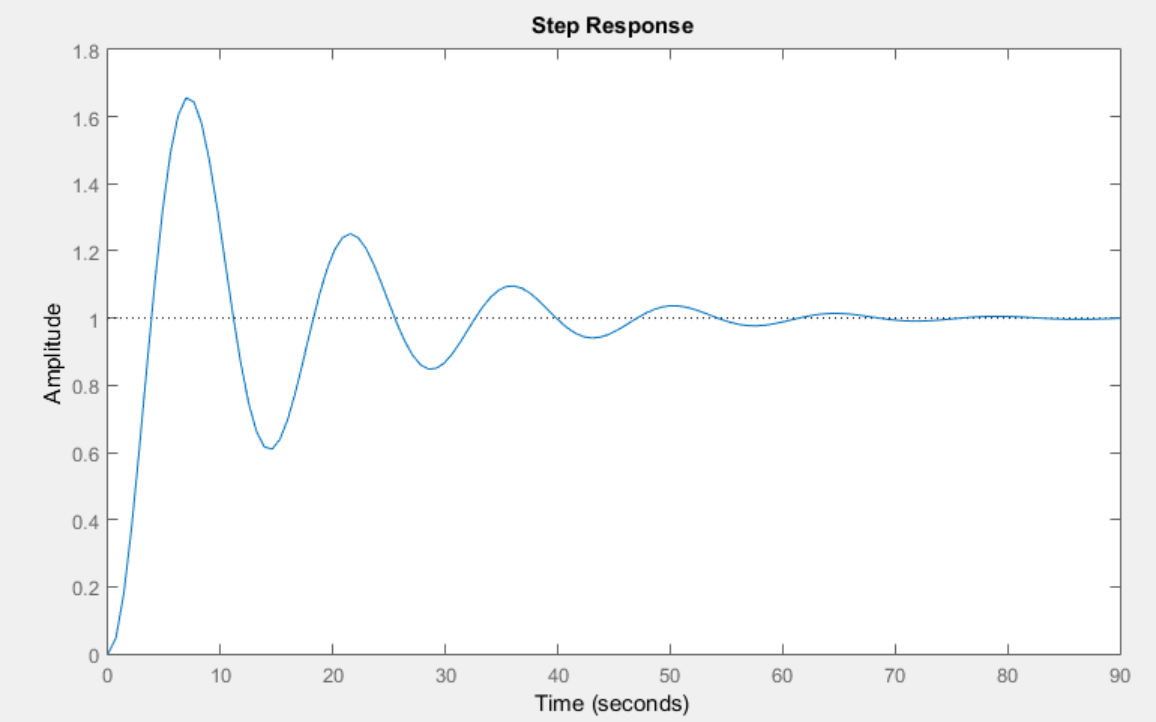
$$A_1 = {a_1}^2 - 2 * a_0 * a_2 \tag{52}$$

$$A_2 = {a_2}^2 - 2 * a_1 * a_3 + 2 * a_0 * a_4 \tag{53}$$

V prostředí Matlab jsem vytvořil skript, který vypočítá podmínku monotónnosti metody optimálního modulu. Ze získaných dvou rovnic o dvou neznámých dalšími úpravami dostaneme výsledný regulátor (54).

$$G_r = \frac{25{,}81 * s + 2{,}791}{s} \tag{54}$$

Přechodová charakteristika uzavřeného obvodu ukazuje překmit 65,6 %. Doba ustálení je 58.7 s. To jsou velmi vysoké hodnoty, ale oproti době ustálení bez regulátoru jsou uspokojivé, zvláště doba ustálení výstupu požadované hodnoty. Podle výsledků z charakteristik (Obr. 21) byl regulátor navržen uspokojivě.

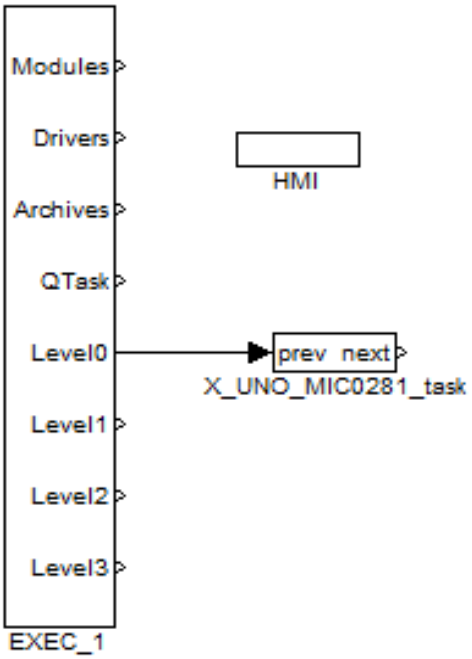


Obr. 21: Přechodová charakteristika uzavřené smyčky s navrženým PI regulátorem metodou optimálního modulu.

3.3 Samotný program v řídicím systému REX

První, co bylo třeba udělat před samotným programováním a použitím řídicího systému REX na platformě REXduino, bylo nastavení obou zařízení pro vzájemnou komunikaci. Na stránkách REXu je ke stažení soubor s připraveným programem pro Arduino UNO, který byl třeba nahrát do mikrokontroleru. Je připravený od samotných tvůrců vývojového prostředí REX a slouží pro zasílání a přijímání stavů a dat do ze zařízení master přes USB rozhraní, které zastupuje Raspberry Pi. Na Raspberry Pi jsem nainstaloval runtime jádro RexCore, na tom běží celý program a řídí jednotlivé naprogramované úlohy. RexCore se nainstaluje přes terminál Linuxu skrze příkazy zaznamenanými v příručce „Začínáme se systémem REX na zařízení Raspberry Pi“, která je ke stažení také na stránkách vývojářů.

Po spuštění REX draw bylo nutné nastavit Executivitu programu (Obr. 22), která taktuje regulační obvod a celý program nastavenou vzorkovací frekvencí. Konfiguruje jednotlivé subsystémy a bloky navázané na něj jsou zpracovány překladačem RexComp pro sestavení celé aplikace.

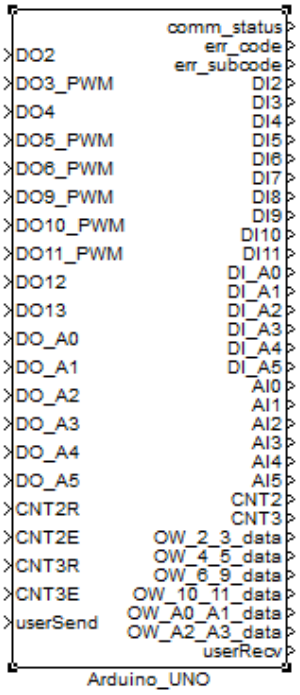


Obr. 22: Nastavení executivy řídícího algoritmu

Dále jsem napojil na Executivu funčkní bloky zvané Task, obsahující samotný algoritmus řízených úloh. Aplikace může obsahovat více úloh. Jak vidět z obrázku (Obr. 22), je třeba systém zkonfigurovat tak, aby seděli z výpočetními úrovněmi level0-level3.

Funkční blok HMI se stará o propojení a generováním vlastní vizualizace řízené úlohy. Tomuto tématu se věnuji v další kapitole, viz 2.4 na str. 20.

Programování v RexDraw se provádí za použití tzv. funkčních bloků. Každý funkční blok má svoji přednastavenou funkci, ale je i možné naprogramovat si funkční bloky vlastní. Nejdůležitější funkčním blokem mé práce je funkční blok Arduino mask (Obr. 23) naprogramovaný přímo od vývojářů systému REX. Má na starost komunikaci se slave zařízením Arduino UNO, na které jsou připojeny senzory, LED diody a jiné ovládací prvky. Po rozkliknutí lze každý vstup nebo výstup nastavit na digitální, analogový, či jiný dostupný způsob čtení nebo zápisu. Funkční blok je stažitelný na stránkách vývojářů a propojováním okolních bloků k jednotlivým vstupům a výstupům se tvoří struktura programu.



Obr. 23: Arduino UNO mask

3.3.1 Popis použitých bloků

- **CNB**

Zadává do řídicího algoritmu konstanty logického datového typu. V programu je zobrazen žlutou barvou a pracuje jako spínání či nastavování ostatních bloků.

- **CNR**

Zadává do řídicího algoritmu konstanty reálného datového typu. V programu je zobrazen modrou barvou a pracuje jako nastavování parametrů v reálném tvaru (desetinná čísla).

- **MP**

Generuje na svém výstupu do řídícího algoritmu pulz o nastavitelné délky. Nastavitelná je také reakce vstupu na náběžnou nebo sestupnou hranu. V programu je zobrazen růžovou barvou a používám jej jako tlačítko bez aretace.

- **NOT_**

Výstupní signál neguje vstupní signál (převrací logickou hodnotu na opačnou).

- **OR_**

Výstupem je logický součet dvou vstupních signálů.

- **RS**

Pracuje jako RS klopný obvod. Přivedeme-li na vstup S signál v hodnotě ON, na výstupu Q se nastaví hodnota ON. Pakliže je na vstup R, který slouží jako resetovací, přivedeme signál ON, výstup Q se změní na hodnotu OFF i v případě, že je vstupní signál S stále v hodnotě ON.

- **From**

Blok slouží pro připojení signálu k jinému bloku v řídícím algoritmu odkazuje se na blok GoTo, který slouží jako zdroj tohoto signálu. Při překladu řídícího algoritmu vyhledává blok GoTo se stejným jménem.

- **GoTo**

Zdroj signálu, který je odkázán do bloku From. Při překladu řídicího algoritmu vyhledává blok From se stejným jménem.

- **LIN**

V řídicím algoritmu počítá lineární interpolaci. Přepočítává vstupní signály u_1 a u_2. na výstupní signály y_1 a y_2.

- **SETPR, SETPB**

Nastavuje vzdálené parametry ostatních bloků v řídicím algoritmu. Poslední písmenko určuje, jedná-li se o parametr datového typu boolean B nebo real R.

- **Display**

Název sám o sobě mluví za vše, slouží k zobrazování hodnoty vstupního signálu u bez použití diagnostiky RexView nebo funkce monitor selection.

- **TRND**

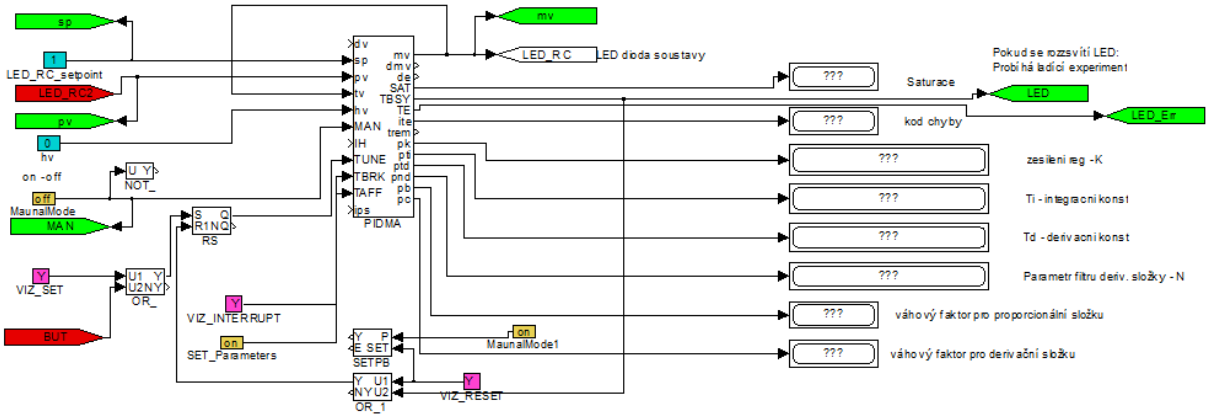
Slouží k ukládání vstupních signálů a zobrazení průběhu signálu. Zaznamenává a zobrazuje až 4 vstupní signály. Zaznamenaná data lze exportovat do formátu csv, se kterými lze pracovat v jiných programech jako je např. Matlab.

- **PIDMA**

  PID regulátor s momentovým autotunerem. Je důležitý pro mnou zvolenou soustavu mé práce, slouží jako PID regulátor s rozšířenou funkcí o automatického nastavení parametrů. V manuálním režimu je nastavená hodnota hv kopírována na výstup akčního zásahu regulátoru. [20]

3.3.2 Popis procesu řídícího algoritmu

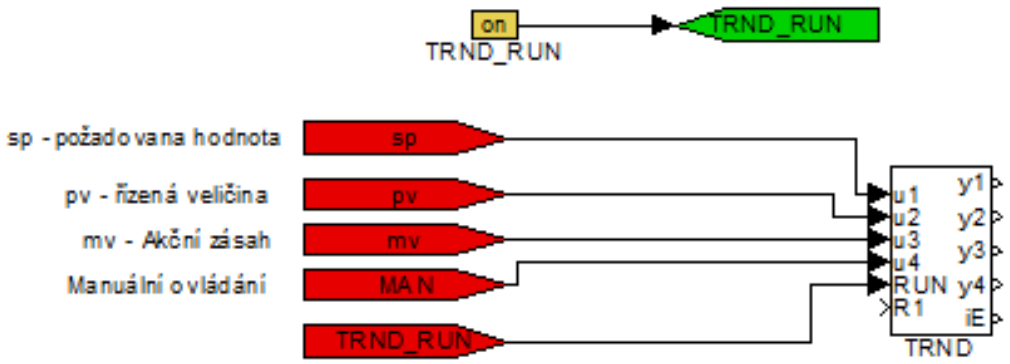
Při spuštění řídicího algoritmu (Obr. 24) nemá blok PIDMA nastavené parametry regulátoru. Ty se nastaví tak, že spustíme ladící experiment. Regulátor nemá nastavené žádné parametry a je třeba, aby byl v ustáleném stavu v pracovním bodě. Nastaví se typ regulátoru, který chceme použít (PI nebo PID) a parametry autotuneru. Před spuštěním autotuneru nastavíme blok požadované hodnoty LED_RC_setpoint, které chceme docílit (v mém případě je to hodnota 1). Musíme mít vypnutý manuální režim. Pokud je manuální režim zapnutý přepneme na režim automatický a zahájíme ladící experiment tlačítkem na výukovém modelu nebo blokem VIZ_SET (slouží pro vizualizaci). Rozsvítí se LED dioda signalizující probíhající proces autotune. Pokud experiment proběhne, LED dioda přestane svítit. Proces ladícího experimentu je viditelný v bloku display, kde se postupně zobrazuje číselný kód. Je-li kód záporné číslo, zobrazuje fázi ladění během experimentu. Zobrazí-li se po ukončení režimu kladné číslo, zelená LED dioda přestane svítit a rozsvítí se bílá LED dioda, došlo k chybě během ladícího experimentu a kód nám označuje ke které chybě došlo. Pokud ladící experiment proběhne správně, bílá LED dioda po zhasnutí zelené LED diody nesvítí a kód chyby je nulový. Po správném ukončení experimentu lze PID parametry automaticky nastavit nebo nechat regulátor nenastavený. O tuto činnost se stará funkční blok SET_parameters. PID parametry nastavuji ihned po ukončení experimentu ladění.



Obr. 24: Model řídícího algoritmu

Ladící experiment lze sledovat i v bloku TRND (Obr. 25), zobrazuje vstupní signály (sp, pv, mv a MAN). Signál sp signalizuje požadovanou hodnotu nastavenou pro automatický režim. Signál pv

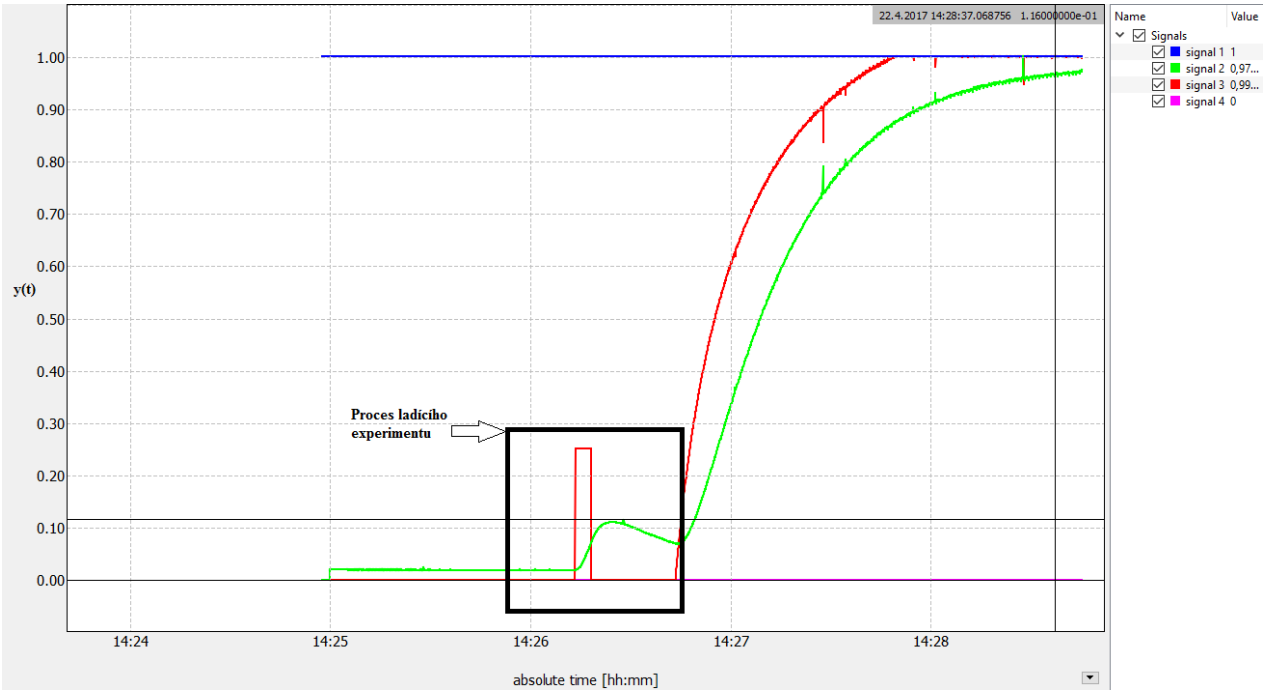
signalizuje řízenou veličinu a signál mv zase akční zásah. Signál MAN zobrazuje přepínání mezi automatickým a manuálním režimem. V bloku TRND lze přesně vidět, co se při ladícím experimentu děje.



Obr. 25: Zapojení bloku TRND

Proces autotune lze přerušit funkčním blokem VIZ_INTERUPT, ten je možné použít pouze ve vizualizaci nebo v prostředí RexDraw (pro nedostatek tlačítek na modelu). Pokud chceme provézt ladící experiment znovu je možné parametry resetovat funkčním blokem VIZ_nulovani. Ten, jako u bloku VIZ_INTERUPT, není možné hardwarově spustit.

Na Obr. 26 vidíme průběh ladícího experimentu. Na začátku procesu se odhaduje drift a šum regulované veličiny (zelený signál) a následně na vstup je aplikován pravoúhlý pulz (červený signál), při kterém dochází k odhadu prvních třech momentů jeho odezvy. Velikost pulsu je nastavitelná parametrem amp funkčního bloku PIDMA. Délka je určena dosažením velikosti regulované veličiny o hodnotu větší, než je nastaveno tolerancí dy. Po ukončení ladícího experimentu je vidět v grafu okamžité aplikování parametrů regulátoru a náběhu regulované veličiny na požadovanou hodnotu.



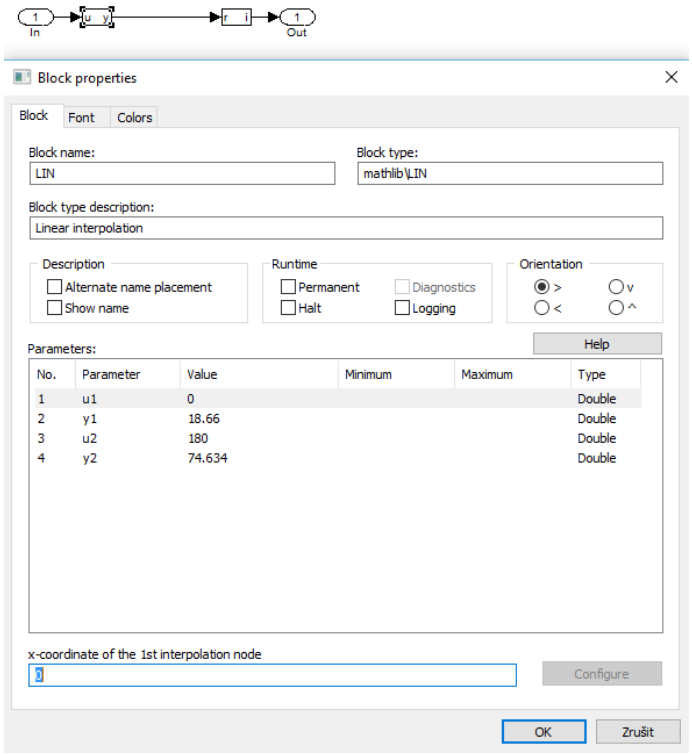
Obr. 26: Proces experimentálního ladění

U možnosti řízení servomotoru jsem použil pulzně šířkové modulace na výstupech Arduina. Pro servomotor HS-311 lze zjistit, že pro otočení doleva potřebuje puls 0,6 ms a pro otočení doprava 2,4 ms při předpokládané periodě 20 ms ve frekvenci 50 Hz. U Arduina lze použít frekvenci 122 Hz a za použití dělitele na výstupu 256 jsem udělal následující přepočty do funkčního bloku lineární interpolace (Obr. 27).

$$T = \frac{50 * 20}{122} \doteq 8{,}2\ s \tag{55}$$

$$y_1 = \frac{0{,}6}{8{,}2} * 255 = 18{,}6 \tag{56}$$

$$y_2 = \frac{2{,}4}{8{,}2} * 255 = 74{,}63 \tag{57}$$

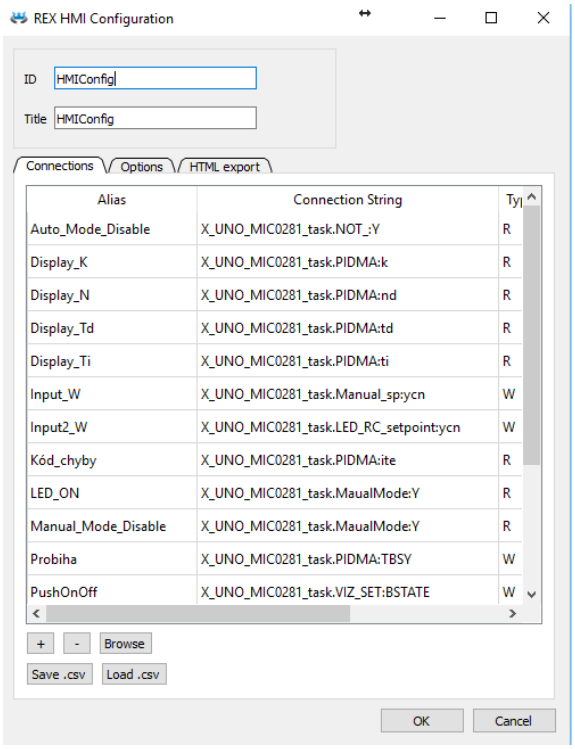


Obr. 27: Nastavení parametrů pro ovládání servomotoru

3.3.3 Vizualizace

Uživatelskou vizualizaci tvoří webová aplikace vytvořená v RexHMI Designeru. To je rozšířený open-source vektorový grafický editor Inkscape. Skládá se z několika vytvořených objektů, které jsou součástí knihovny programu jako například LED dioda, posuvník, real-time graph. Lze vytvořit i vlastní objekty, ale pro mou práci byly vyhovující stávající objekty. Objekty se propojují s vytvořeným řídícím algoritmem. Pro spojení je důležité, aby byla spuštěná exekutiva programu. Pak se lze nastavením IP adresy k REXduinu připojit a vyčíst signály, které chceme ovládat nebo sledovat.

Připojení provedeme tak, že v programu otevřeme dialogové okno Edit Element a zobrazí se seznam objektů, které lze propojit. Klikneme-li na tlačítko Browse a připojení je v pořádku zobrazí se strom programu s možnými body připojení. Všechny objekty mají svou položku Connection String (Obr. 28). Do této kolonky je možné přiřadit požadovaný přípojný bod. Kolonka Type složí k určení zápisu nebo čtení zvoleného přípojného bodu. V záložce HTML export zvolíme cestu, kam se konečná vizualizace bude generovat a ke které s řídící algoritmus bude připojovat.



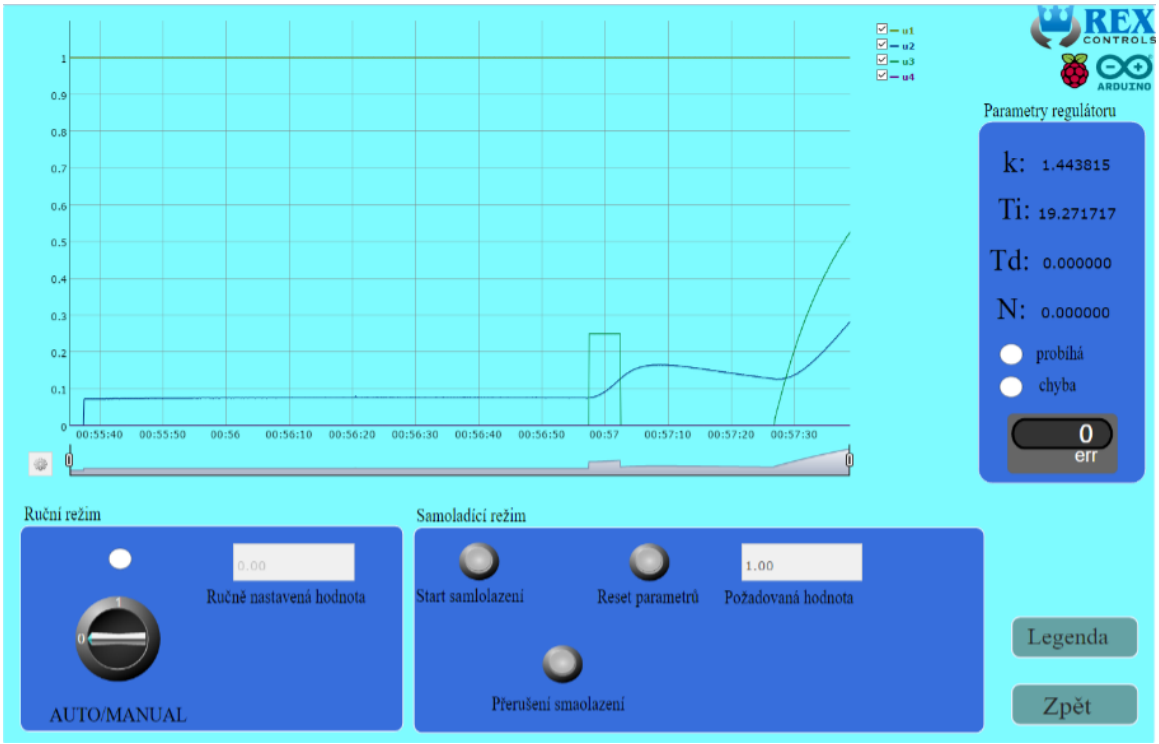
Obr. 28: Navázání vizualizačních proměnných na signály řídícího algoritmu

Vytvořil jsem 4 obrazovky, jedna složí jako úvodní obrazovka menu (Obr. 29), která slouží jako rozcestník k dalším obrazovkám. Další dvě jsou určeny k přímému ovládání regulátoru PIDMA a servomotoru. A čtvrtá funguje jako nápověda k obrazovce ovládající regulátor PIDMA.

Obr. 29: Úvodní obrazovka

Obrazovka regulátoru PIDMA (Obr. 30), se skládá z objektu TRND, Dsiplay, PushOnOff, SwitchOnOff, ControlLed, Input. Označením objektu a stisknutím klávesové zkratky Ctrl + E dojde k zobrazení dialogového okna Edit element, zde můžeme nastavit objekt, jak potřebujeme a v kolonce value zapisujeme název bodu propojení.

Přepínačem AUTO/MANUAL volíme samoladící režim nebo režim manuální, kde na výstupu regulátoru je kopírovaná hodnota zadaná v poli „Ručně nastavená hodnota“. V tomto režimu svítí kontrolka nad přepínačem. V samoladícím režimu tlačítkem „Start samolazení“ začne proces ladícího experimentu, tlačítkem „Přerušení samolazení“ dojde k přerušení procesu. Tlačítko „Reset parametrů“ resetuje nastavené parametry PID regulátoru a proces samolazení můžeme opakovat. Po správném nastavení parametrů se parametry znázorní na obrazovce pod svými proměnnými. Proces samolazení signalizuje kontrolka „probíhá“. Dojde-li k chybě kontrola „chyba“ se rozsvítí a pod ní se znázorní kód chyby. Ten je možné pomocí tlačítka „Legenda“ porovnat v jiné obrazovce a chybu identifikovat.



Obr. 30: Obrazovka PIDMA

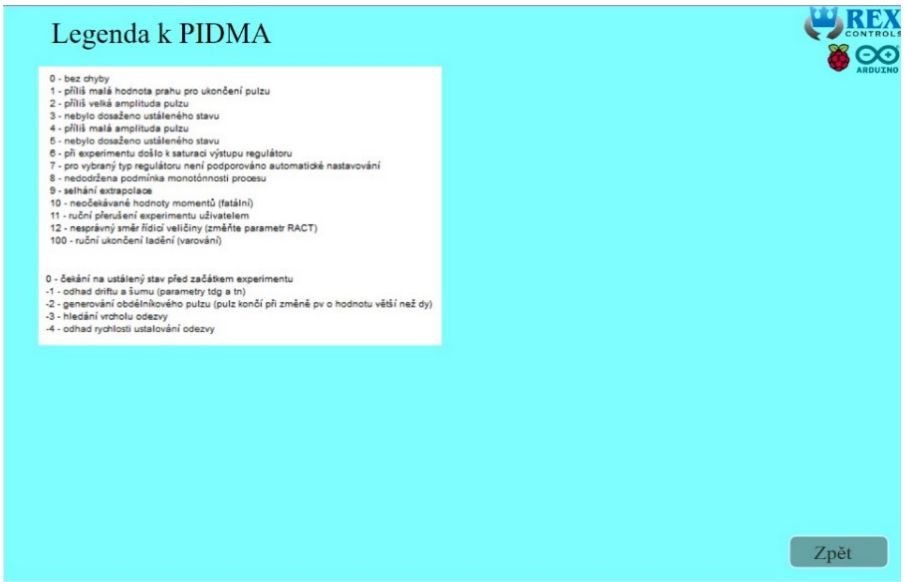
Objekt TRND vykresluje graf a je propojený k funkčnímu bloku TRND v řídicím algoritmu. Je možné v něm signály zaškrtnutím zobrazit nebo naopak schovat. Objekt Display slouží k zobrazování hodnot, zaznamenané v řídicím algoritmu, jako jsou parametry nebo aktuální proměnné. PushOnOff pracuje jako tlačítko bez aretace a je napojené na funkční bloky generující pulzní impuls v řídicím algoritmu. ControlLed zobrazuje binární stav On nebo Off. Objekt Input

slouží k zápisu číselných hodnot do řídicího algoritmu v rozsahu 0 - 100 a v závislosti na zvoleném režimu (samoladícím nebo manuálním) je možné do něj zapisovat.

Obrazovka Servo (Obr. 31) ovládá servomotoru a zobrazuje aktuální hodnotu natočení servomotoru. Složena je z objektu SliderHorizontal a objektu Gauge180. Posuvník slouží k nastavování polohy servomotoru v rozmezí 0-180° a na zobrazovači výchylka ručičky znázorňuje natočení polohy.

Obr. 31: Obrazovka servo

Obrazovka legenda (Obr. 32) je pouze pro zobrazení informací k ovládání regulátoru PIDMA.



Obr. 32: Obrazovka legenda

4 Závěr a zhodnocení výsledků

Cílem práce bylo navrhnout a realizovat testovací stojan se zapůjčeným výukovým modelem, na kterém byla vybrána fyzikální úloha druhého řádu a na ní aplikován regulační obvod pomocí řídicího systému REX. Testovací stojan slouží pro demonstraci laboratorních úloh předmětu Kybernetika a byl zkompletován z plastového stojanu, na kterém byla připevněna platforma REXduino, vytvořená propojením Raspberry Pi s Arduinem UNO. To znamenalo seznámit se s použitou platformou a vytvořením řídícího algoritmu ve funkčních blocích řídicího systému REX. Dále byla realizována vizualizace, díky které by mohl uživatel celý experiment ovládat přes webový prohlížeč.

V teoretické části jsem se věnoval popisu mikrokontroléru Arduino UNO s mini počítačem Raspberry Pi 3, které jsou součástí platformy REXduino. Věnoval jsem se popisu řídicího systému REX, sloužícího k realizaci řídicích algoritmů automatického řízení. Dále jsem se zaměřil na problematiku identifikace soustavy a návrhu regulátorů PID.

V praktické části jsem se zaměřil na návrh regulátoru pro zvolenou soustavu druhého řádu. Abych mohl provézt návrh regulátoru, musel jsem získat představu, jak se daná soustava chová. Zvolil jsem proto experimentální identifikaci systému a vybudil jsem měřenou soustavu jednotkovým skokem. Na výstupu jsem pak změřil její výstupní hodnoty a v programu Matlab jsem naměřená data identifikoval pomocí Identification Toolbox. Identifikovaný přenos druhého řádu odpovídá z 98,92 % naměřeným datům. Porovnal jsem identifikovaná data s daty provedené analytickou identifikací z teoretické části (kapitola 2.6) a zjistil, že se až na menší odchylky téměř shodují.

Z matematicko-fyzikálního modelu získaného prvotní identifikací jsem provedl syntézu regulačního obvodu. Pro názornou ukázku jsem si vybral dvě metody, metodu Ziegler-nicholsovu a metodu nastavení optimálního modulu. Porovnal jsem mezi sebou získané parametry obou metod a zjistil jsem, že obě metody vykazují vysoký překmit regulační odchylky, ale kratší dobu ustálení oproti době ustálení bez použití regulátoru.

Následně jsem vytvořil řídící algoritmu, ve kterém jsem použil funkční bloku PIDMA, realizující PID regulátor s automatickým laděním. V tomto kroku nastal problém, protože tento blok není součástí bezplatné licence řídicího systému REX. Proto po dohodě s vývojáři mi byla časově bezplatně zapůjčena. Po dokončení samoladícího experimentu, regulátor nastavil parametry s přesností na dvě desetinná čísla k = 1,44, Ti = 19,27 s a Td = 0 s. Derivační složka nebyla v regulaci použita. Použitím řídícího algoritmu s funkčním blokem PIDMA, jsem docílil rychlejšího a efektivnějšího návrhu regulátoru, který je v dnešní době v praxi běžně používán.

Na konec jsem vytvořil uživatelskou vizualizaci, kterou by mohl operátor celý experiment ovládat. Pro přehlednost jsem vytvořil více obrazovek odkazující se na úvodní obrazovku. Tím jsem docílil

oddělením od jednotlivých aplikačních řešení řídícího algoritmu a tím i zpřehledněním aplikace. Použil jsem integrovaného RexHMI designeru, z důvodu možností přímého propojením se signály řídícího algoritmu. Je však možné i na napojení SCADA systému jako je Reliance, Iconics či Wonderware InTouch, za použití OPC DA serveru REX.

Testovací stojan s výukovým modelem bude sloužit k výuce předmětu Kybernetika, kde bude mít za cíl seznámit studenty s návrhem regulačního obvodu regulované soustavy druhého řádu a seznámením se s řídicím systémem REX. Je možné aplikovat řídicí systém i na ostatní fyzikální úlohy, kterými výukový model disponuje nebo použít řídicí systém REX na řízení domovní automatizace, řízení otáček motoru, řízení manipulátorů či řízení jiných složitějších úloh.

5 Literatura

[1] *Santy* [online]. Savana.cz [cit. 2017-04-24]. Dostupné z: http://www.santy.cz/arduino-c2/arduino-uno-r3-nano-shield-atmel-328p-i47/

[2] *Arduino* [online]. 2014 [cit. 2017-04-09]. Dostupné z: https://arduino.cz

[3] VODA, Zbyšek. *Průvodce světem Arduina* [online]. HW kitchen. [cit. 2017-04-09].

[4] *Arduino UNO* [online]. Trademarks, 2017 [cit. 2017-04-09]. Dostupné z: http://www.arduino.org/products/boards/arduino-uno

[5] *Opensource: What is a Raspberry Pi?* [online]. [cit. 2017-04-11]. Dostupné z: https://opensource.com/resources/what-raspberry-pi

[6] *Raspberry Pi* [online]. [cit. 2017-04-11]. Dostupné z: https://www.raspberrypi.org

[7] *Arduino-shop.cz* [online]. ECLIPSERA [cit. 2017-04-24]. Dostupné z: http://arduino-shop.cz/arduino/1385-raspberry-pi-3-model-b-quad-core-1-2-ghz-64bit-cpu-1gb-ram-wifi-bluetooth-4-1-1473147122.html

[8] *My electronics lab: Raspberry Pi 3 Pinout Model B, RPI2, B+ 40Pin GPIO Pinout* [online]. satya sankar sahoo, 2016 [cit. 2017-04-24]. Dostupné z: https://www.myelectronicslab.com/tutorial/raspberry-pi-3-gpio-model-b-block-pinout/

[9] REX CONTROLS S.R.O. *Začínáme se systémem REX na platformě Raspberry Pi* [online]. Plzeň, 2016, 29 s. [cit. 2017-04-11]. Verze 2.10.8. Referenční číslo dokumentace: 5898.

[10] *Co je to řídící systém REX?* [online]. Plzeň: pixlab.cz, 2000 [cit. 2017-04-24]. Dostupné z: https://www.rexcontrols.cz/rex

[11] *Dokonalé spojení Raspberry Pi a mikrokontroléru Arduino* [online]. Plzeň: pixlab.cz, 2014 [cit. 2017-04-24]. Dostupné z: https://www.rexcontrols.cz/clanky/dokonale-spojeni-raspberry-pi-a-mikrokontroleru-ar

[12] *Základy počítačové grafiky* [online]. 2011 [cit. 2017-04-24]. Dostupné z: http://moodle.zshk.cz/mod/page/view.php?id=3548

[13] *RexHMI – A Web-based HMI for REX: User guide*. Version 2.50.1. Plzeň (Pilsen), Czech Republic: REX Controls, 2016.

[14] *HMI pro automatizaci bazénu pomocí Raspberry Pi* [online]. Plzeň: pixlab.cz, 2014 [cit. 2017-04-24]. Dostupné z: https://www.rexcontrols.cz/clanky/hmi-pro-automatizaci-bazenu-pomoci-raspberry-pi

[15] NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. Ostrava: Montanex, 1999. ISBN 80-7225-030-2.

[16] OŽANA, Štěpán. *Navrhování a realizace regulátorů: učební text*. Ostrava: Vysoká škola báňská - Technická univerzita, 2012. ISBN 978-80-248-2605-9.

[17] METODY SEŘIZOVÁNÍ PID REGULÁTORŮ. Pardubice, 2015. Bakalářská práce. Univerzita Pardubice. Vedoucí práce Ing. Libor Kupka, Ph.D.

[18] *Zieglerova-Nicholsova metoda kritických parametrů* [online]. [cit. 2017-04-24]. Dostupné z: http://www.352.vsb.cz/uc_texty/synteza/text0302.htm

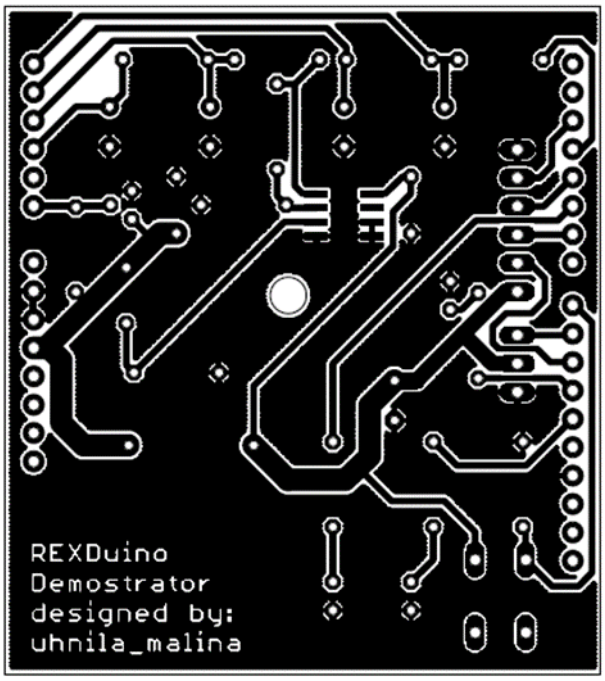
[19] *Metoda optimálního modulu* [online]. [cit. 2017-04-24]. Dostupné z: http://www.352.vsb.cz/uc_texty/synteza/text0305.htm

[20] *Funkční bloky systému REX* [online]. Verze 2.10.8. Plzeň: REX controls, 2016 [cit. 2017-04-24]. Dostupné z: https://www.rexcontrols.cz/media/HTML/DOC/CZECH/index.html

[21] ŠIMČÍK, Petr. *Inkscape – Praktický průvodce tvorbou vektorové grafiky*. COMPUTER PRESS, 2013, 296 s.

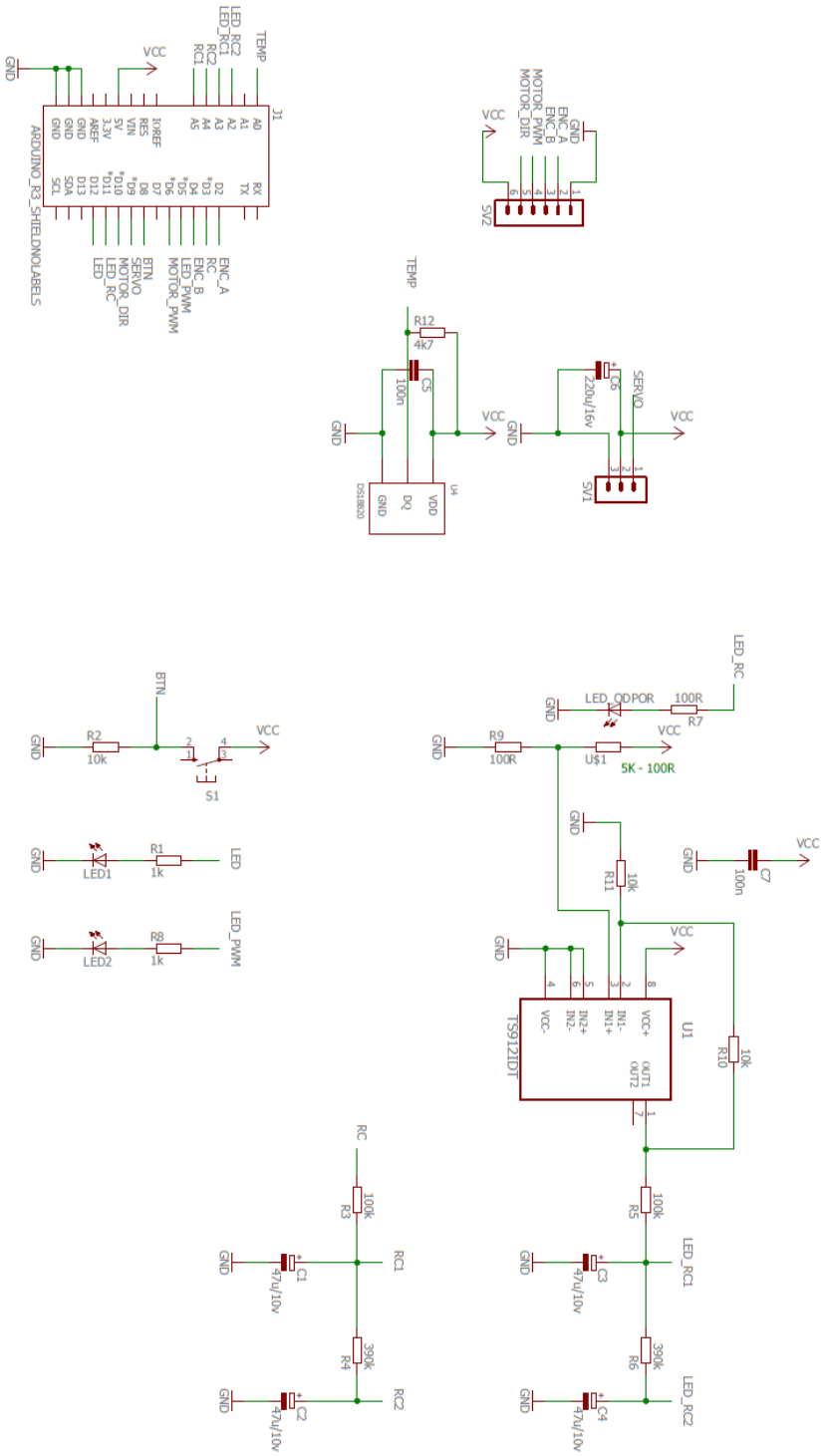
6 Seznam příloh

Příloha I – Plošný spoj modelu



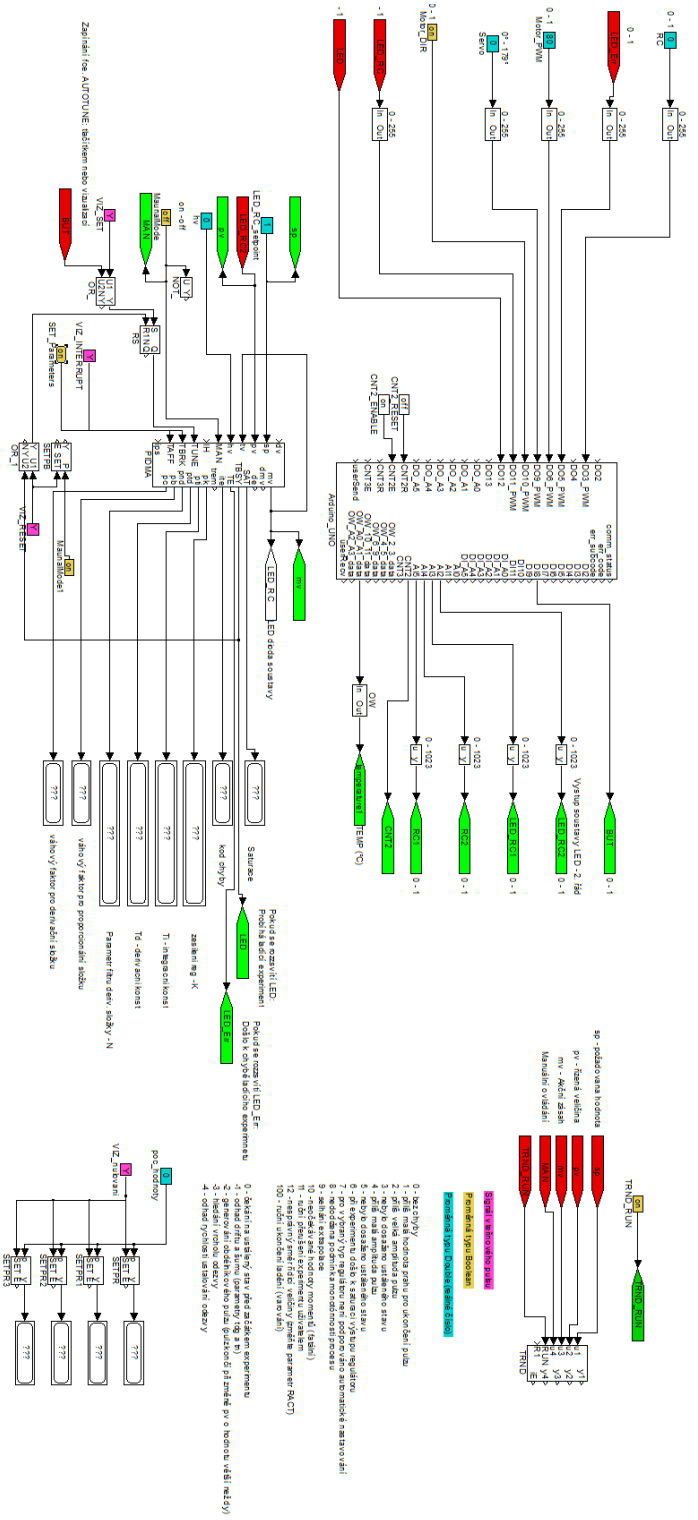
Obr. 33: Plošný spoj výukového modelu

Příloha II – Elektronické schéma jednotlivých fyzikálních úloh



Obr. 34: Elektronické schéma jednotlivých fyzikálních úloh

Příloha III – Řídící algoritmus řízené úlohy



Obr. 35: Řídící algoritmus fyzikální úlohy

Příloha IV – Obsah CD

Bakalářská práce v elektronickém formátu PDF

Obrázky – složka s obrázky testovacího stojanu, vizualizace a řídícího algoritmu

Skripty pro Matlab – složka se skripty pro Matlab

Model pro identifikaci jednotkového skoku – řídící algoritmus v Rex controls pro identifikaci systému

Model řídícího algoritmu BP – hlavní řídící algoritmus v Rex controls

REXduino_slave.ino – program pro komunikaci s mastrem Raspberry Pi

REX_Getting_Started_RasPi_CZ.pdf – uživatelská příručka